REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 9 Jan 97	3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
An Image Fusion Algorithm For Spatially Enhancing Spectral Mixture Maps

5. FUNDING NUMBERS

6. AUTHOR(S)
Harry N. Gross

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rochester Institute of Technology

8. PERFORMING ORGANIZATION REPORT NUMBER

96-46D

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
DEPARTMENT OF THE AIR FORCE
AFIT/CIA
2950 P STREET
WPAFB OH 45433-7765

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

19970117 118

14. SUBJECT TERMS

15. NUMBER OF PAGES
118

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94

AN IMAGE FUSION ALGORITHM FOR SPATIALLY ENHANCING SPECTRAL MIXTURE MAPS
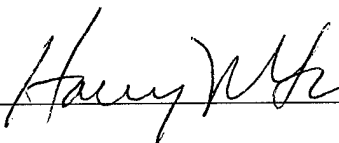
by

Harry N. Gross
Major, USAF

B.S. United States Air Force Academy (1983)
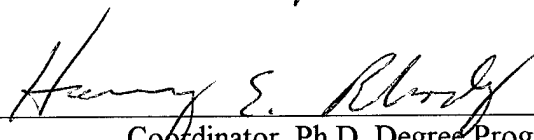S.M. Massachusetts Institute of Technology (1985)

A dissertation submitted in partial fulfillment
of the requirements for the degree of Ph.D.
in the Chester F. Carlson Center for Imaging
Science in the College of Science of the
Rochester Institute of Technology

August 1996

Signature of the Author _____

Accepted by _____
Coordinator, Ph.D. Degree Program

CHESTER F. CARLSON
CENTER FOR IMAGING SCIENCE
COLLEGE OF SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL
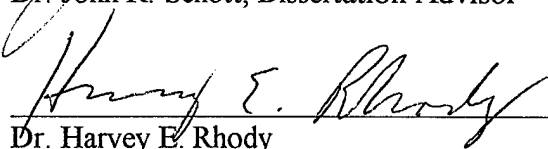
Ph.D. DEGREE DISSERTATION
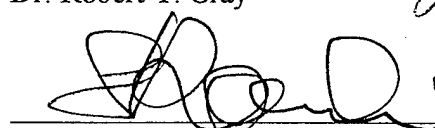
The Ph.D. Degree Dissertation of Harry N. Gross
has been examined and approved by the dissertation
committee as satisfactory for the dissertation
requirement for the Ph.D. degree in Imaging Science

Dr. John R. Schott, Dissertation Advisor

Dr. Harvey E. Rhody

Dr. Robert T. Gray

Dr. Stanislaw P. Radziszowski

August 14/96
Date

DISSERTATION RELEASE PERMISSION
ROCHESTER INSTITUTE OF TECHNOLOGY
COLLEGE OF SCIENCE
CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE

Title of Thesis: An Image Fusion Algorithm for Spatially Enhancing Spectral Mixture Maps

I, Harry N. Gross, hereby grant permission to the Wallace memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Signature: _____

Date: _14 Aug 96_____

An Image Fusion Algorithm for Spatially Enhancing Spectral Mixture Maps

by

Harry N. Gross

ABSTRACT

An image fusion algorithm, based upon spectral mixture analysis, is presented. The algorithm combines low spatial resolution multi/hyperspectral data with high spatial resolution sharpening image(s) to create high resolution material maps. Spectral (un)mixing estimates the percentage of each material (called endmembers) within each low resolution pixel. The outputs of unmixing are endmember fraction images (material maps) at the spatial resolution of the multispectral system. This research includes developing an improved unmixing algorithm based upon stepwise regression. In the second stage of the process, the unmixing solution is sharpened with data from another sensor to generate high resolution material maps. Sharpening is implemented as a nonlinear optimization using the same type of model as unmixing.

Quantifiable results are obtained through the use of synthetically generated imagery. Without synthetic images, a large amount of ground truth would be required in order to measure the accuracy of the material maps. Multiple band sharpening is easily accommodated by the algorithm, and the results are demonstrated at multiple scales. The analysis includes an examination of the effects of constraints and texture variation on the material maps. The results show stepwise unmixing is an improvement over traditional unmixing algorithms. The results also indicate sharpening improves the material maps.

The motivation for this research is to take advantage of the next generation of multi/hyperspectral sensors. Although the hyperspectral images will be of modest to low resolution, fusing them with high resolution sharpening images will produce a higher spatial resolution land cover or material map.

Acknowledgments

I am indebted to my advisor, John Schott, and the other members of my dissertation committee. Your guidance at each step of this research was gratefully accepted. Each member of the committee contributed a different perspective to the problems at hand. I thank you for helping me maintain an aggressive research schedule.

I also recognize the indirect contributions of my fellow students. Thank you for showing interest in my research, listening as I tried to understand, and offering advice when I needed it. I counted on my fellow Air Force officers, the other Ph.D. students, and anyone who happened to walk into the computer lab. I particularly thank the DIRS staff and students for their assistance. You showed me how to get things done.

Finally, I thank my family. The email connection made each day more interesting and gave me a mechanism for quick technical support. Most importantly, I thank Amy, Lauren, and Bradley. Your love, understanding, and support give me strength always.

Table of Contents

List Of Figures

List of Tables

1. Introduction

1.1 Image Fusion

Analysts use remote sensing images to gain information about a target or land area that cannot be obtained by direct measurement. They use the information in the images to infer characteristics of the objects in question. For example, analysts may be interested in crop health, land use, or mapping. The particular objects may have been imaged many times, with different sensors. Clearly, an analyst interested in the most accurate description would want to include as many images as possible as part of the "evidence" that is examined. Using data from all available sensors in the study would enable a thorough analysis.

The data available may include images taken from both satellites and aircraft. The actual sensor platform, of course, affects the image characteristics. However, knowing these characteristics, the analyst can account for any differences in the acquisition parameters. The specific sensors may also differ. For example, the detector materials (which dictate the underlying spectral sensitivity) may not be the same. The combination of a detector material with a filter or a diffraction grating defines the spectral response. The detector size, optical path characteristics, and the sensor altitude combine to determine the ground sample distance corresponding to an image pixel. This is the spatial response.

Image fusion merges images of different spatial and spectral resolutions to create a high spatial resolution multispectral combination. Spatial resolution is the size of a pixel projected onto the ground. Spectral resolution corresponds to the spectral width of the detector/filter in the sensor.

Many image fusion algorithms combine the various images at the digital count level. The result is a set of multispectral, high spatial resolution images. However, these images must often be further processed to create maps of the materials in the scene. This dissertation presents an image fusion algorithm that directly produces high resolution material maps.

1.2 Spatial vs. Spectral Resolution

With passive sensors, the detector simply measures the incident energy. This energy, represented by the radiance at the detector, is a function of the radiance from different sources. The various types of radiance include energy that is reflected from the target, self-emitted from the target, as well as background and atmospheric energy. A large part of remote sensing is modeling all these radiance terms in order to estimate features of the target.

At the sensor, the digital counts for the image are typically taken as a linear function of the detected radiance,

$$dc = gain \cdot \iiint radiance \cdot dA \cdot d\Omega \cdot d\lambda + bias. \tag{1}$$

Equation (1) shows the sensor will integrate the energy within a differential area, solid angle, and wavelength. The differential area, dA, is a pixel. The differential solid angle, $d\Omega$, is the projection of the pixel, through the optical path, to the ground. The combination dA $d\Omega$ is the spatial resolution. The differential wavelength, $d\lambda$, corresponds to the spectral bandwidth, and is the spectral resolution. In order to have confidence in the measured radiance, the signal to noise ratio (SNR) at the detector must be adequate. This requirement on SNR is what determines the detector size, and results in the tradeoff between spatial and spectral resolution.

As illustrated in Figure 1, if a narrow filter is used to give a high spectral resolution, the amount of electromagnetic radiation that makes it through the filter will consequently be small. Because of the relatively small number of photons, the detector size must be made large in order to maintain SNR. The large detector size, when projected through the sensor optics, results in a large spot size on the ground. This is a low spatial resolution.

2
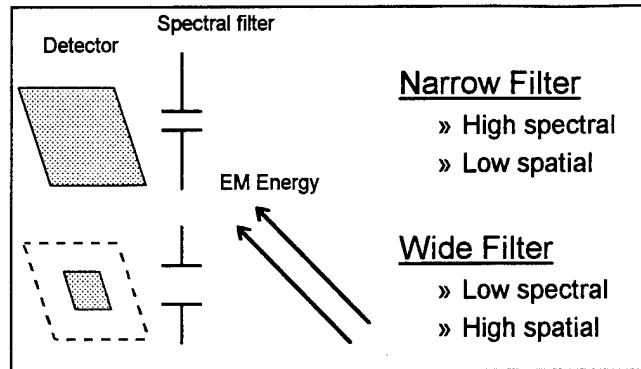
Figure 1: Spectral vs. Spatial Resolution

Conversely, if a wide spectral filter is used, we have low spectral resolution, but a large number of photons can reach the detector. Therefore, the detector can be smaller and, thus, the ground spot is small. A low spectral resolution implies a high spatial resolution.

This tradeoff between spectral and spatial resolution means that analysts will always be presented with a variety of image data with which to perform image fusion.

One may consider any scene as having a reflectance which can be written as a function of spatial location and wavelength, e.g., $r(x,y,\lambda)$. Image data is often represented as a cube. The face of the cube shows how the objects in the scene vary spatially – an "image." The depth of the cube corresponds to different wavelengths. Slices of the cube represent image bands. While the underlying data for the cube are continuous, the sensor samples the cube in x and y to make pixels, and in depth to form bands. Figure 2 shows an image cube. Because of the perspective view, the spectral response of the materials along the top and side is visible. The bright band shows the vegetation reflectance is greatest in the near infrared.
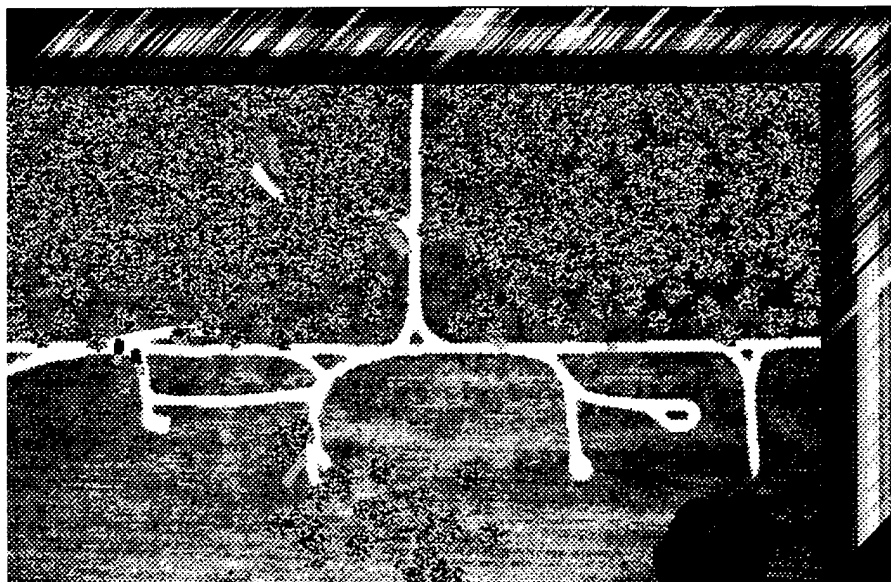
Figure 2: Image Cube

Sensors sample this imaging space with different characteristics. For the image fusion problem, data from at least two different sensors are merged. The sensor with lower spatial resolution, but higher spectral resolution is the *spectral* sensor. The sensor with high spatial resolution but low spectral resolution is the *spatial* sensor. The spectral sensor image cube typically will have many slices, or spectral bands, to offset its poor spatial resolution. Those slices would be thinner than the slice(s) from the spatial sensor. On the other hand, a spatial sensor would have more data values (pixels) in each band than a corresponding area from the spectral sensor.

Typically, sensors with a few spectral bands are called *multispectral*. Their bandwidths are usually on the order of 100 nm. *Hyperspectral* sensors imply dozens or more narrow (i.e. 10 nm) spectral bands. *Panchromatic* refers to image bands with several hundred nm bandwidths. In image fusion applications, panchromatic data is usually used to spatially "sharpen" multi or hyperspectral image data. For this work, the distinction between multispectral and hyperspectral is not usually relevant, and both terms are used interchangeably. Clearly, some applications need hyperspectral sensors with medium to low resolution. Other applications require high spatial resolution panchromatic data. Image fusion uses both these data sources to create a hybrid image.

Imagery comes from a variety of sources. Two commonly available commercial satellites are Landsat and SPOT. Landsat is a series of satellites launched by the United States in the 1970s and 1980s. The latest vehicles, numbers 4 and 5, were launched in 1982 and 1984 respectively. A large library of Landsat images exists. One sensor on Landsat 4 and 5 is called the Thematic Mapper (TM) and has seven spectral bands. Six of the bands have 30 meter spatial resolution and contain data in visible and infrared spectral regions. The seventh band (actually band number six) gives thermal information at a 120 meter pixel size.

The Systeme Pour l'Observation del la Terre (SPOT) is a French satellite system that has launched three vehicles in 1986, 1990, and 1993. SPOT has 3 spectral bands in the visible and NIR region with 20 meter pixels. It also has a panchromatic band with a 10 meter resolution. One example of image fusion is to combine the Landsat 30 meter spectral data with a SPOT 10 meter panchromatic image. Another, almost trivial, fusion challenge is to combine SPOT spectral bands with the SPOT panchromatic.

Future sensors will improve the resolution in different ways. One class of sensors will have higher spatial resolution. Several companies are planning high resolution commercial satellites with only a few spectral bands (typically visible or near-infrared). They will have resolutions of a few meters. Russian, Canadian, and Japanese satellites will also provide commercially available imagery (Foley, 1994).

At the other extreme are imaging spectrometers such as AVIRIS and MODIS. The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) is a NASA sensor which flies on an ER-1 (U-2) aircraft. The sensor has 224, 10 nm spectral bands at wavelengths from 0.4 to 2.5 μm. The nominal spatial resolution is 20 meters (Vane et al., 1993, Johnson and Green, 1995). NASA is planning to launch the Moderate-Resolution Imaging Spectroradiometer (MODIS) sensor in the late 1990's. This satellite, a part of the Earth Observing System program, will have low spatial resolution (250-1000 meters) with 36 spectral bands covering wavelengths from 0.4 to 14.5 μm (NASA, 1995). Future fusion possibilities are combining AVIRIS with digitized air-photos or high resolution commercial satellite data, or increasing the MODIS resolution to the level of Landsat TM.

1.3 Correlation

Fusion is possible because of the large amount of correlation in images. The bands of multispectral images tend to have some degree of correlation. (This is one way the data can be compressed.) Also, the panchromatic image band spectrum typically overlaps with some of the spectral bands. This band correlation is very important in determining how well the fusion algorithms work.

1.3.1 Band Correlation

An analysis of TM data (e.g., for compression purposes) shows there are less than seven bands worth of information in the data. Digital counts in any given band are correlated with digital counts in others. In fact, Crist and Cicone (1984) estimate the TM data can largely be represented in 3 or 4 dimensions. For typical scene objects, knowing digital counts in one band makes the digital counts in another fairly predictable.

Figure 3 shows normalized reflective TM and SPOT panchromatic bands and their overlapping sensitivities (Markham and Barker, 1985). Clearly in the overlapping regions, the digital counts will be highly correlated. The SPOT panchromatic band is highly correlated with TM bands 2 and 3.
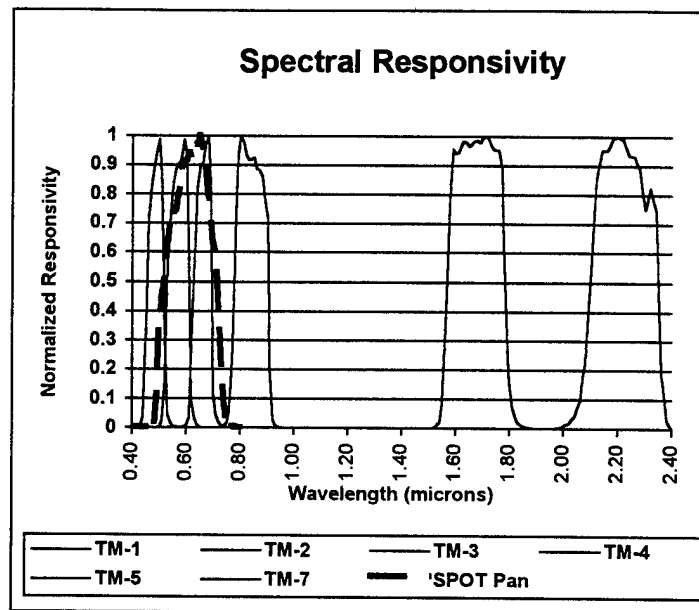
Figure 3:   Spectral Responsivity of TM and SPOT Panchromatic Bands

Because of this correlation, it would be easy to fuse SPOT panchromatic with TM visible bands. However, using SPOT to improve the TM infrared bands (especially bands 5 and 7) is more problematic. Image fusion algorithms address this by creating models to predict how the high resolution data will appear in bands that correspond to the spectral sensor. The various methods for accomplishing this will be reviewed in the next chapter.

1.3.2 Material Correlation

An alternate way to capitalize on the correlation within an image is to use material reflectance curves. Spectral Mixture Analysis models the total radiance measured at the sensor as a linear combination of radiance (reflectance) from a number of materials (Adams et al., 1986, Smith et al., 1990a, Adams et al., 1993). Spectral unmixing is the process that takes digital counts and calculates the percentage of each material within the pixel. If the materials in the scene are identified, their reflectance curves could be used as a basis for mapping between bands.
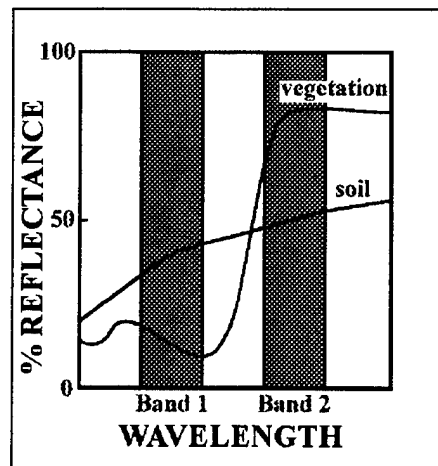
Figure 4:   Material Correlation

Figure 4 shows a sketch of two material reflectance curves. If the material were known, one could predict its reflectance in alternate bands. In this example, even though vegetation is darker than soil in band 1, identifying the materials predicts a brighter vegetation contribution in band 2.

7

1.4 Mixed Pixels

The region on the ground imaged by a single pixel may contain a variety of materials. In some respects, the types of materials depend upon the application. For example, a pixel may be 100% forest, or classified as a mixture of deciduous and coniferous trees. Farmland may be classified as agricultural vs. urban, corn vs. wheat, or healthy vs. stressed. By making the material classes more specific, almost all pixels become "mixed." Conversely, if the classes are general, many of the pixels may be "pure." However, even with general classes, pixels that lie along the boundaries will encompass more than one material. A *mixed pixel* is a pixel containing more than one type of material of interest. Obviously, whether or not a pixel is mixed depends upon the application.

One can envision many kinds of material combinations. For convenience, distinctions will be made among three types of mixtures. An *intrinsic* mixture is defined as one whose constituent materials interact at a microscopic level. Photons striking intrinsic mixtures may encounter multiple interactions. Thus, the average reflectance is likely to be a complex combination of the individual material properties. Unmixing intrinsic mixtures requires a nonlinear model and is not addressed in this work.

Aggregate and *areal* mixtures, on the other hand, are characterized by linear interactions between the materials and incoming photons. They consist of distinct materials, but are mixed at various spatial scales. Aggregate mixtures combine on a macroscopic level. The total reflectance is a spatial average of the constituent materials, but their components are not spatially separable at the sensor resolution. Areal mixtures also combine linearly, but their components are spatially resolvable, especially with a high resolution sensor. The three types of mixtures are illustrated in Figure 5.

8
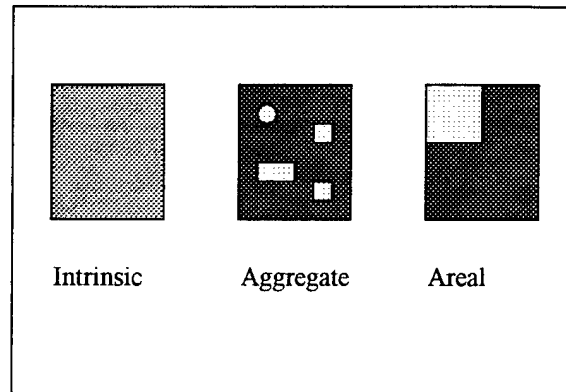
Figure 5: Basic Types of Mixtures

Envision a linear mixture which is not spatially resolvable by the low resolution spectral sensor. To this sensor, the material is an aggregate mixture. One could unmix the pixel to determine subpixel composition, but could not spatially locate the endmembers. However, to a high spatial resolution sensor, the mixture could be areal. The second sensor could be used to sharpen the material maps obtained from the low resolution unmixing.

Spatially separating areal mixtures is the motivation for developing image fusion algorithms.

1.5 End Result

1.5.1 High Resolution Digital Counts

One goal of image fusion is to merge a Low (spatial) Resolution Multi-Spectral (LRXS) set of images with a High (spatial) Resolution Panchromatic (HRP) image to create a High Resolution Multi-Spectral (HRXS) hybrid. Figure 6 is a graphical depiction of the fusion concept.
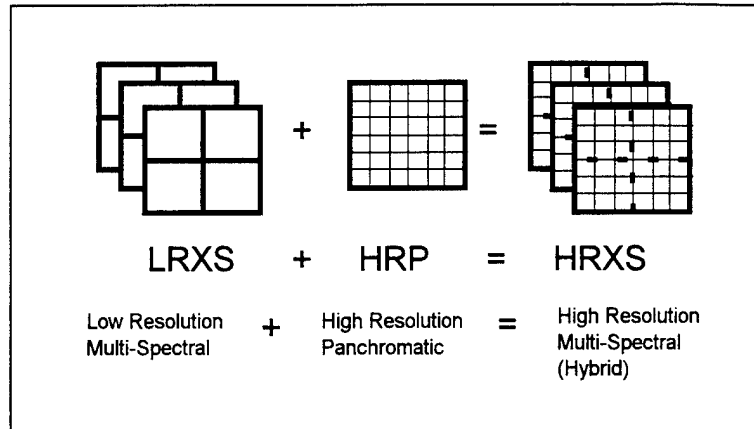
Figure 6: Image Fusion Concept – Combining Digital Counts

The large pixels of the LRXS image are called *superpixels*. Assuming the images are properly registered, a superpixel corresponds to a number of smaller *subpixels* in the HRP image.

Since the images come from different sensors, there will, in general, not be a simple correspondence between the pixels in the two images. However, fusion algorithms depend on accurately registering the separate images. This requires accounting for distortions due to differing acquisition parameters, as well as rescaling and interpolating to account for the different pixel sizes. This project does not examine the effects of registration. It assumes that registration is accomplished with very small error. For example, pixel oversampling via interpolation allows registration to be done to subpixel accuracy using interactive control point selection.

Fusion at the digital count level works best for strongly correlated bands. In weakly correlated bands, the performance deteriorates. It is especially difficult to fuse mixed pixels in weakly correlated bands.

1.5.2 High Resolution Material Maps

Alternatively, the analyst frequently desires material maps of the area. One purpose of image fusion is to improve classification performance. The hypothesis is that the HRXS image would give better classification results than just using the HRP image. Classifying directly from the LRXS data would obviously give low resolution results.

The fusion procedure proposed in this study applies spectral unmixing to the LRXS images to derive material maps. Then, the HRP images are used to *sharpen* the material maps to a higher resolution. The end product is an image (map) for each material. The intensity in these images is made proportional to the fraction of the material present. Figure 7 illustrates fraction images.
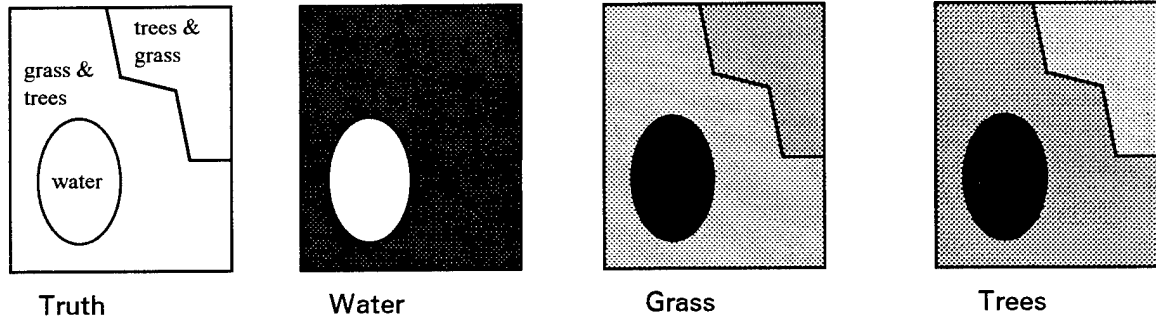
Figure 7: Notional Fraction Images (Material Maps)

1.6 Synthetic Imagery

Since this proposed fusion algorithm aims to create high resolution material maps, quantifying the algorithm performance is difficult. In the absence of a great deal of ground truth data, material maps are often evaluated on anecdotal evidence.

To ensure accurate knowledge of the underlying materials during algorithm development and testing, synthetic image generation (SIG) tools are used to create the imagery. SIG controls all the image parameters, making it easier to analyze algorithm performance. It also aids algorithm development by regulating variation. The image fusion algorithm is developed incrementally by including increasing amounts of realism in the synthetic imagery. The nature and degree of error is observed at each stage of development, and adjustments made to the algorithm to reduce the errors. Algorithm design with SIG imagery is useful to measure and improve even widely accepted algorithms like spectral unmixing, where quantitative performance has been difficult to document.

A SIG developed image can be used to control the various error sources that are likely to impair the algorithm performance. These error sources include atmospheric effects, mismodeled spectral

endmembers, and variability in topography and illumination. By controlling the introduction of these errors, the robustness of the algorithm can be studied and improved upon.

The algorithm development work capitalizes on two benefits of synthetic image generation (SIG). SIG scenes can be built with varying degrees of complexity. The ability to incrementally increase realism gives feedback to algorithm designers. They may test their designs under increasingly difficult conditions, and modify the designs to incrementally improve robustness. Secondly, since SIG is entirely computer created from a defined data source, the underlying "truth" is known. Algorithms can be evaluated under various conditions, where their performance can be quantified and compared to alternate techniques. Only after the algorithm is shown to work under simulated conditions is it then tested on real imagery.

1.7 Outline

The objective of this research was to develop an alternate image fusion algorithm based upon spectral unmixing. Spectral unmixing transforms hyperspectral data from the image domain to material maps. To date, spectral unmixing products have only been generated at low spatial resolution. Fusing the material maps with high resolution sharpening image(s) yields a more spatially accurate classification map. Quantifiable results are available because SIG imagery is used.

A secondary objective was to improve the spectral unmixing algorithm. Traditional unmixing calculates material maps for an entire scene. However, the same materials are not present in all the pixels within the image. An algorithm is presented which selects the materials to be unmixed on a pixel by pixel basis.

This document is organized as follows. Section two provides background reference on alternate image fusion techniques. These would establish a baseline for comparing image fusion performance. Section three briefly describes the proposed algorithm as a combination of unmixing and sharpening. The fourth section contains the mathematical foundation required to design and develop code to implement the algorithm. Section five contains quantified results using synthetic test imagery. The results show sharpening the material maps reduces the error in the material fractions. The high resolution maps are a

more accurate representation of the ground truth. The last section summarizes the contributions made by this research and lists some recommendations for future study.

2. Alternate Fusion Techniques

2.1 Image Fusion Paradigm

The growth in remote sensing is a relatively recent phenomenon. Only in the last decade or so have the data become generally available and affordable. In addition, advances in computer technology have just recently placed the required digital image processing power in low cost workstations. As a result, many of the image processing tools are new and not well tested. The algorithms have been applied to only a small number of images. As the literature shows, sometimes the results are scene or image dependent (Braun, 1992). Sensor designs continue to improve, and the trend is towards better resolution, both spatial and spectral. All these factors combine to create a rapidly changing field with many creative and successful ideas.

Image fusion combines images of different spatial and spectral resolutions to make a multispectral combination. The most difficult aspect of image fusion is accounting for the different spectral responses in the bands to be fused. Figure 8 is a block diagram representation of the generic fusion process. Two steps are involved. First, the sensor bands are aligned using a transformation. Second, the data are combined in some manner. In some methods, an inverse transformation is required as a third step. Using this paradigm, one can review the existing fusion techniques.
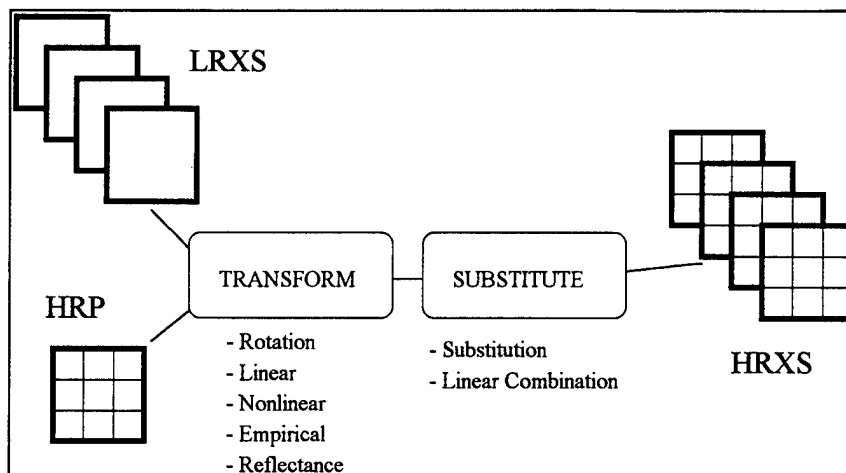
Figure 8:   General Image Fusion Process

2.1.1 Transformations

Several types of transformations are used in the literature. The goal of the transformation stage is to account for the different spectral sensitivities of the low resolution and panchromatic bands. Some algorithms, called Component Substitution (COS), use a *coordinate transformation* to rotate the multispectral data so that one of the new axes lies in the same direction as the panchromatic band. Shettigara (1992) presents a generalized COS technique.

Another popular transformation is a *linear regression*. Price (1987), Munechika et al. (1993), and Braun (1992) use linear regression models to predict multispectral digital counts as functions of the panchromatic and other spectral bands. These high resolution estimates are used in the ratio method technique.

Multiresolution decomposition techniques like wavelets are used by Ranchin et al. (1993), Iverson and Lersch (1994), and Ranchin and Wald (1996) to establish relationships between panchromatic and multispectral data at various resolutions. These data points are used as training data for a *nonlinear model* that relates the bands.

Munechika (1990) created an *empirical model* with a Lowtran simulation and sensor models for his data. He ran his simulation for many conditions and used an ensemble average transformation.

15

Finally, this project proposes to use spectral mixture analysis as the transformation model. Spectral mixing (or unmixing) generates estimates of the fractions of materials in each pixel using the material *reflectance* curves (Smith et al., 1990a). Since the average spectral response of each material is known, it is easy to transform from one spectral band to another. Furthermore, since the transformation is done according to the objects in the scene, contrast reversals in uncorrelated bands are recognized.

2.1.2 Combinations

After the spectral bands are transformed, the data must be combined. The literature contains only a few combination methods. The COS algorithms use a substitution of panchromatic data for one of the transformed spectral bands. Most of the other methods use a linear combination of multispectral and panchromatic data. A wavelet reconstruction can be done by applying the transformation to the detail in the image (the wavelet) rather than the entire image. In this methodology, the wavelets are orthogonal while filtered versions of the panchromatic image are correlated.

The following section summarizes the most common fusion algorithms.

2.2 Modeling Band Correlation

Munechika (1990) distinguishes three classes of fusion algorithms. The first class is called "fusion for visual display." These algorithms are primarily concerned with making an image that looks good to a human interpreter. Simple histogram manipulation and contrast stretching may fit into this category. These methods are easy, and tend to give reasonable results, which explains why image fusion is so popular. They require no transformation other than scaling. One example is to substitute the high resolution panchromatic data into the computer CRT display green channel. Multispectral red and blue are left unchanged. Since the human visual system is most sensitive to green, this gives a pleasing result.

The second class is termed "fusion by separate manipulation of the spatial information." These are the COS algorithms. In these techniques, the high resolution panchromatic data is assumed to lie in a particular direction in a specified image space. The multispectral data are transformed into that image space. Part of the data is replaced with the panchromatic data. The new image set is then transformed

back into the original spatial domain. Chavez et al. (1991) and Braun (1992) compared three algorithms of this class.

2.2.1 Coordinate Transformations

2.2.1.1 High Pass Filter (HPF)

The High Pass Filter (HPF) is the most obvious way to separately manipulate the spatial data. Schowengerdt (1980) suggests an image can be represented as the sum of a lowpass filtered image and a highpass filtered image, i.e.,

$$PAN = L_{PAN} + H_{PAN} \qquad (2)$$

If the high resolution data contains the edges not visible in the low resolution set, this technique may be used to replace those missing edges. Schowengerdt's HPF uses the corresponding multispectral data for the low pass image,

$$HRXS = LRXS + K \cdot H_{PAN} \qquad (3)$$

where K is selected to appropriately weight the combination of low resolution multispectral and filtered panchromatic images. Filberti et al. (1994) uses HPF to fuse color aerial photography with AVIRIS hyperspectral imagery.

2.2.1.2 Intensity Hue Saturation (IHS)

The Intensity Hue Saturation (IHS) technique is described in both Chavez et al. (1991) and Braun (1992). Some other recent applications are described in Carper et al. (1990) and Ehlers (1991). The IHS technique can only be used for three bands of data. The transformation is similar to a color space manipulation. The three low resolution bands of data are treated as colors (for example, red, green, and blue). Thus, they can equivalently be represented by an intensity, a hue, and a saturation. Intensity is similar to lightness. It would have a scale from black to white. The hue is the dominant color, which would consist of blues, greens, yellows, reds, and purples. The saturation is the amount of that color. The saturation scale would run from gray to fully colored. The three-band spectral data are transformed into

17

IHS space. The intensity image is removed, and replaced with the scaled panchromatic image. This hybrid is then transformed back to RGB. The IHS algorithm assumes the panchromatic and intensity images are similar.

2.2.1.3 Principal Components Analysis (PCA)

The Principal Components Analysis (PCA) is a well-known transformation in Linear Algebra and is used in many control system formulations. The transformation takes a vector of correlated data and changes it into orthogonal components. These components are uncorrelated with each other. Richards (1986) uses PCA to analyze multispectral images. In the image fusion application, PCA is used as a COS algorithm to separate the first principal component. This first component should contain the data that are common to all the bands, and is likely to be similar to the panchromatic image. First, the multispectral data are transformed into principal component space. The first principal component image is removed and replaced with the scaled panchromatic image. The hybrid is then transformed back into multispectral space. Image fusion using PCA is described in Chavez et al. (1991), Braun (1992), and Shettigara (1992).

2.2.2 Multiresolution Decomposition

Multiresolution decomposition can be used to develop a relationship between the panchromatic and spectral data. For example, an orthonormal wavelet decomposition is done on both images to generate resolution (Laplacian) pyramids. The levels of these pyramids represent subsampled detail images that are uncorrelated across the respective bands. The panchromatic image has one extra layer in the pyramid due to its higher spatial resolution. A nonlinear model relates the panchromatic and spectral digital counts at each of the subsampled layers. Once this model is trained, it is applied to the high resolution panchromatic image to predict a high resolution spectral image. Ranchin et al. (1993) and Iverson and Lersch (1994) use this method to fuse SPOT multispectral and SPOT panchromatic. The 2:1 resolution ratio is especially suited to multiresolution decomposition. Ranchin and Wald (1996) use the method to improve SPOT spectral data by a factor of ten.

2.2.3 Ratio Methods

An unfortunate characteristic of COS techniques is the appearance of effects of one multispectral band into another band due to the coordinate transformation. Therefore, Munechika has labeled a third category "fusion for radiometric integrity." With these algorithms, prime importance has been given to properly allocating each band's digital counts in the hybrid image. Several authors have recognized that computer segmentation algorithms will perform further operations on the digital counts. Therefore, radiometric accuracy is critical. This third category of techniques is based upon the ratio method.

The *ratio method* is a straightforward approach to parceling the low resolution energy into high resolution pixels. It works best on bands that are highly correlated. One simple ratio method is attributed to Pradines (1986). He uses the following equation to merge the SPOT spectral bands with the SPOT panchromatic band:

$$HRXS = LRXS \frac{HRP}{\displaystyle\sum_{superpixel} HRP} \tag{4}$$

where *HRXS* is the desired High Resolution Multi-Spectral digital count, *LRXS* is the digital count from the Low Resolution Multi-Spectral superpixel, and *HRP* is the digital count from the High Resolution Panchromatic subpixel. Recall a subpixel refers to the small pixels in the high resolution image. A superpixel corresponds to a collection of subpixels that is equivalent in size to the low resolution pixels. Pradines does no band transformation as his method was designed to fuse the first two SPOT multispectral bands with the highly correlated SPOT panchromatic band.

Price (1987) proposes a two stage process. He uses a ratio for the strongly correlated bands and a Look-Up Table (LUT) for the weakly correlated bands. His ratio equation is similar to Pradines'

$$HRXS_i = LRXS_i \frac{HRXS'}{HRXS'_{avg,i}} \tag{5}$$

Instead of directly using *HRP*, Price uses a regression routine to estimate the high resolution data. His model is

$$LRXS_i = a_i PAN_{avg} + b_i \tag{6}$$

The low resolution data and the averaged PAN data are used at the coarse resolution to derive a set of regression coefficients. Those coefficients are used at high resolution to predict the $HRXS'$ term,

$$HRXS_i^{'} = a_i PAN + b_i. \tag{7}$$

This linear regression compensates for the spectral differences between the PAN and low resolution bands.

Interestingly, Filberti et al. (1994) show that with proper choice of K, their HPF (3) is very similar to Price's ratio method (5). Substituting,

$$K = \frac{LRXS}{L_{PAN}} \tag{8}$$

into (3) gives

$$
\begin{aligned}
HRXS &= LRXS + \frac{LRXS}{L_{PAN}} H_{PAN} \\
&= LRXS \left[1 + \frac{1}{L_{PAN}} H_{PAN} \right] \\
&= \frac{LRXS}{L_{PAN}} \left[L_{PAN} + H_{PAN} \right] \\
HRXS &= \frac{LRXS}{L_{PAN}} PAN
\end{aligned}
\tag{9}
$$

If the bands are highly correlated, Price's $HRXS'$ will be very similar to PAN. Furthermore, when Price averages over the superpixel, it is equivalent to taking a low pass filtered version of the PAN image.

Before discussing Price's LUT for the weakly correlated bands, we can show the modifications made by Munechika (1990) to the ratio equation. Munechika constructs a low resolution synthetic panchromatic band as

$$SYNPAN = \sum_{bands} \psi_i LRXS_i \tag{10}$$

He uses simulations to derive the coefficients $\{\psi\}$ for combining TM bands 1-4 with the SPOT panchromatic band. Munechika ran a LOWTRAN atmospheric model for 5 reflectance curves (objects) in

each of 5 scenes in 3 different atmospheric conditions. These 75 simulation results were combined with the sensor models to create a set of coefficients that best describe the bands' spectral relationship. Munechika uses an empirical model for his transformation. Munechika's ratio equation is

$$HRXS_i = \frac{LRXS_i}{SYNPAN} HRP \tag{11}$$

Munechika makes another modification. Define the ratio

$$R = \frac{LRXS}{SYNPAN} \tag{12}$$

Typically, he calculates the hybrid digital count as

$$HRXS = R \cdot HRP \tag{13}$$

where R is the ratio for the particular superpixel. However, Munechika notes in a mixed pixel, some of the subpixels are more likely to be similar to a neighboring superpixel than to the current one. He proceeds as follows. Compare the $HRXS$ digital count to the $SYNPAN$ (or PAN average) digital count in the current and adjacent superpixels. Identify the superpixel with the value closest to $HRXS$. Use the ratio from that superpixel in equation (13). In this approach, the hybrid average is no longer guaranteed to equal the $LRXS$. Munechika found trading this loss of radiometric integrity improved classification performance.

All these ratio methods are similar. Pradines uses a sum rather than an average. Price uses an unweighted estimate of the data. Munechika creates a synthetic pan band by empirically modeling the specific sensor relationships. His $SYNPAN$ includes a regression weighted by band. Munechika also modifies the ratios for mixed pixels. Braun (1992) found the Price and Munechika algorithms had similar performance.

Developing a high resolution hybrid image in the weakly correlated bands is much more difficult. The $LRXS$ and HRP images are not linearly related.

Price (1987) suggests a Look-Up Table (LUT) approach. The table relates the PAN digital counts to the $LRXS$ digital counts. For each value of PAN avg (0-255 for TM 8-bit data), Price records each value of $LRXS$ digital count that occurs throughout the image. He then averages the values in each bin to create a mapping from PAN avg digital counts to $Average$ $LRXS$ digital counts. Price then enters the LUT with

the *HRP* digital count and extracts a digital count for *HRXS'*. This value is used in his ratio equation (5).

Figure 9 shows an example look-up table.

Pan avg DC	LRXS$_i$ DC	Avg LRXS$_i$ DC
0	8,8,7,9	8
1	20,24	22
2	16,14,15	15
...	...	...
255	...	...

Figure 9: Example Look-Up Table for Fusing Weakly Correlated Bands

Munechika et al. (1993) use a different approach to solve for the weakly correlated bands. They postulate a linear relationship between the weakly correlated bands, the panchromatic band, and the previously predicted bands. At low resolution, digital counts from a neighborhood of 6 superpixels are used to calculate the coefficients in the following regression

$$LRXS_m = a_0 + a_1 SYNPAN + a_2 LRXS_i + ...$$ (14)

where *m* refers to band *m* (weakly correlated) and *i* refers to band *i* (strongly correlated and previously predicted).

These regression coefficients {*a*} are calculated using each of the previously predicted bands. If the regression error is larger than a threshold, another term is added to the equation

$$LRXS_m = a_0 + a_1 SYNPAN + a_2 LRXS_i + a_3 LRXS_j + ...$$ (15)

where bands i and j are both previously predicted bands.

Additional terms (bands) are added until the regression error is reduced below the threshold. Adding terms until the error criterion is satisfied is why the technique is called *extended regression*. Once the error criterion is satisfied, the coefficients are used with the high resolution data to predict the hybrid image

$$HRXS_m = a_0 + a_1 HRP + a_2 HRXS_i + ...$$ (16)

22

where *HRXS$_i$* is known via the ratio method (11) for the strongly correlated band.

One problem with extended regression is it tends to give noisy results when applied to uniform areas of an image. Braun (1992) describes a modification called *global regression*. The global regression algorithm recognizes that local neighborhoods may not be the proper domain in which to perform the regression. Instead, the regression is done with all pixels that contain the same class of ground cover.

First an unsupervised classifier uses the low resolution multispectral and the high resolution panchromatic images to make a class map of all the pixels in the image. All pixels are classified as grass, trees, water, urban, etc. When the regression equation (15) is applied, only pixels that belong to the same ground cover class are used to calculate the coefficients. But, those pixels are not restricted to be in the local neighborhood; they can come from anywhere in the image. Thus the name global regression refers to the regression being performed on pixels spaced globally throughout the image. As before, previously predicted bands are incrementally added to the regression equation until the resultant error decreases below a threshold. Once the coefficients are determined at low resolution, they are applied to the high resolution data with equation (16). Both ratio methods and their extended and global regression modifications are sound fusion algorithms. They use a linear transformation and a ratio to combine the data.

2.3 Algorithm Summary

Chavez et al. (1991) compare three algorithms that separately manipulate the spatial information. They find the Intensity-Hue-Saturation transformation gives poor results since the panchromatic band does not correspond well to the intensity image. Principal Components Analysis works better since the panchromatic image is more similar to the first principal component. Chavez et al. find the High Pass Filter algorithm works best.

Braun (1992) finds the algorithms that maintain radiometric integrity work better than those that separately manipulate the spatial data. The ratio method is the foundation of this approach. Braun finds both the Price (1987) and Munechika (1990) algorithms give similar results in the correlated bands.

Braun finds extended regression works best overall when scenes contain high frequency data (urban areas). The global regression works best when the scene is low or medium frequency (agricultural areas).

In summary, image fusion works well estimating bands that are strongly correlated with the panchromatic data. The ratio method combined with global regression gives the best overall performance. The results are best for scenes with medium and high spatial frequency content.

Fusion does not work as well on images with low frequency information in the weakly correlated bands. The Intensity Hue Saturation algorithm is the most restrictive since it can only be applied to three bands of data. On average, the High Pass Filter and Principal Components techniques do not perform as well as the ratio method.

Algorithms for image fusion work reasonably well. They can predict the high resolution spectral response for many pixels. The performance degrades, however, when predicting weakly correlated bands or when the pixels are mixed. To address these shortcomings, the algorithms add complexity. The next section presents an alternate model based upon spectral mixture analysis. Recasting image fusion as a spectral mixing problem uses the material reflectance curves to ensure all bands are properly predicted. The limitations of spectral mixing are addressed via a priori knowledge, constraining the possible solution set, and proper algorithm design. The next section describes a framework for applying spectral mixture analysis to image fusion.

3. Proposed Algorithm

Typical image fusion algorithms generate high resolution images of digital counts in the various spectral bands. However, digital counts do not reveal what kind of object is in the scene. Spectral unmixing maps the objects, but to date has only been done at low resolution. The desired output product is often a high resolution material map. Such a product would identify materials, and locate them to high accuracy. The contribution of this research is a method of using spectral mixing tools at high spatial resolution. By integrating spectral mixing and image fusion, a high resolution material map is attained.

The proposed image fusion algorithm takes advantage of the spectral correlation between materials by first identifying the materials via spectral mixture analysis. This results in low spatial resolution material maps. These maps are then sharpened via a nonlinear optimization algorithm (Gross and Schott, 1996a). This section summarizes the procedure.

The overall image fusion algorithm is a two stage process depicted in Figure 10. A set of multi/hyperspectral images is represented by an image cube. Spectral unmixing operates on the images in the cube to create a set of material maps at the (low) spatial resolution of the multi/hyperspectral images. Image fusion is accomplished by using one or more sharpening bands to increase the spatial resolution of the material maps.
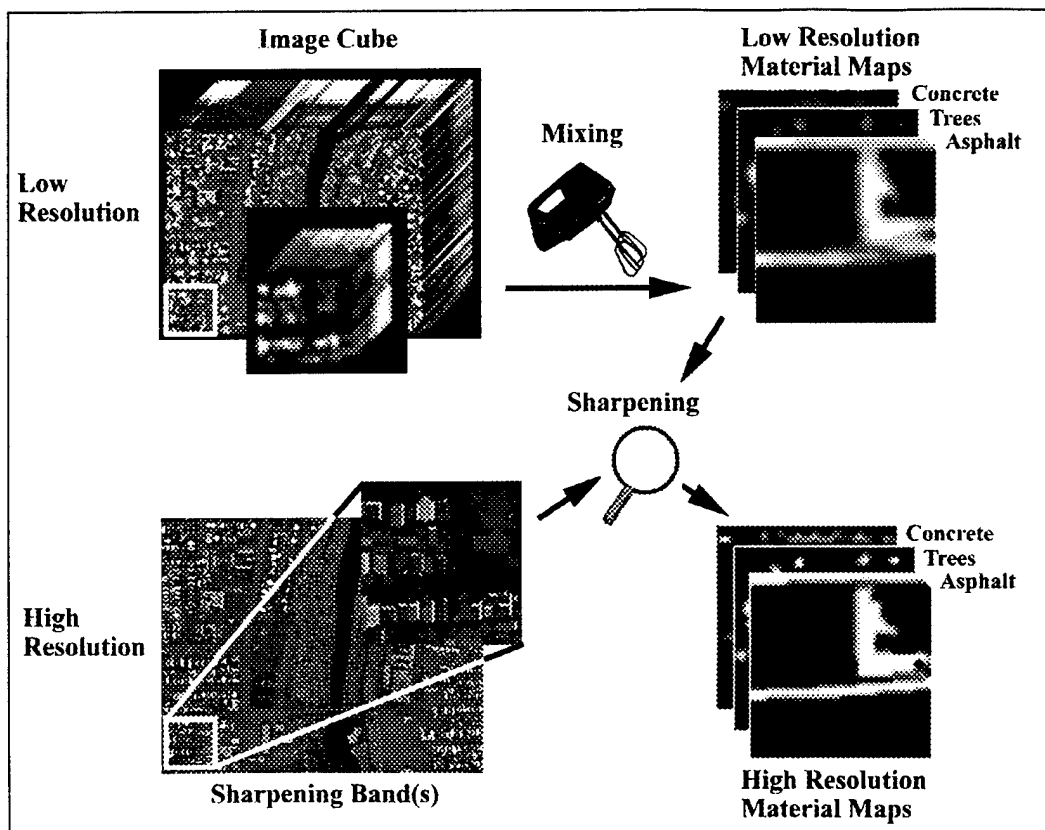
Figure 10: Image Fusion Data Flow – Creating High Resolution Material Maps

3.1 Spectral Mixture Analysis

Spectral Mixture Analysis (Adams et al., 1986, Smith et al., 1990a, Adams et al., 1993) was

developed to address the problem with classifying mixed pixels. The authors recognized the large scale of

low resolution images creates situations where the measured digital counts are due to many materials

within the corresponding ground areas. Multispectral and hyperspectral imaging sensors provide the

ability to extract spectral signatures of individual and mixtures of materials.

The first step in the algorithm generates material maps from the spectral sensor data. The

spectral mixture algorithm transforms digital counts, or radiance, into fractions of the constituent

materials, called endmembers. In many ways, spectral mixture analysis is similar to principal components

analysis. "A key difference is the spectral mixture analysis defines a fixed reference (endmember) that is

spatially and temporally invariant, whereas the principal components vary depending upon the scene,"

(Mertes et al., 1993). The primary outputs of spectral mixture analysis are "fraction images" in which the intensity can be made proportional to the percentage of that particular endmember. This gives a spatial mapping of the endmembers.

As the mathematics will show, the spectral mixture analysis algorithm relies on having more spectral bands than endmembers. Therefore, this technique has been successful in analyzing multispectral or hyperspectral images. To date, however, it has only been applied to images with high spectral resolution, but low spatial resolution. Spectral mixing has only been used to make low resolution material maps.

Spectral mixture analysis has been used with many different sensors and has mapped many different materials. Thematic Mapper (TM) data are used to map desert vegetation (Smith et al., 1990b) and to evaluate sediment in surface waters (Mertes et al., 1993). AVIRIS data are used to separate green vegetation, nonphotosynthetic vegetation, and soils (Roberts et al., 1993), to distinguish soil, grass, and bedrock (Mustard, 1993), and to calibrate apparent surface reflectance (Farrand et al., 1994). Spectral mixture analysis of TM data and synthetic aperture radar data is used to separate vegetation from rock (Evans and Smith, 1991). Analysis of thermal infrared images is used to evaluate desert rock types (Gillespie, 1992). Spectral mixture analysis is also used to determine the optical components of inland tropical waters (Novo and Shimabukuro, 1994). Finally, image processing and linear systems analysis are used to improve spectral mixing fraction images (Wu and Schowengerdt, 1993).

The mathematical model for spectral mixing is straightforward. In a particular pixel, for the ith spectral band, let

$$dc_i = gain_i \cdot radiance_i + bias_i \tag{17}$$

where dc are digital counts recorded by the detector, $gain$ and $bias$ are band dependent sensor values, and $radiance$ is the effective radiance at the sensor. Note that radiance is affected by the detector's responsivity, $\beta(\lambda)$,

$$radiance_i = \int_\lambda radiance(\lambda)\beta_i(\lambda)d\lambda \tag{18}$$

27

Spectral mixture analysis assumes the radiance reaching the detector can be modeled as a linear sum of radiance due to n different endmembers

$$radiance(\lambda) = \sum_{e=1}^{n} L_e(\lambda) f_e \tag{19}$$

where $L_e(\lambda)$ is the spectral radiance of the particular endmember and f_e is the fraction of the eth endmember in a pixel. Combining,

$$radiance_i = \int_\lambda \sum_{e=1}^{n} L_e(\lambda) f_e \beta_i(\lambda) d\lambda$$

$$= \sum_{e=1}^{n} \int_\lambda L_e(\lambda) \beta_i(\lambda) d\lambda f_e$$

$$radiance_i = \sum_{e=1}^{n} L_{i,e} f_e \tag{20}$$

where $L_{i,e}$ is the effective radiance of the eth endmember seen by the ith spectral band of the detector.

The terms $gain_i$ and $bias_i$ account for sensor effects. Typically, imaging systems use preflight or onboard calibration to calculate these linear correction terms. A gain and bias are reported by band along with the image data. Assuming the terms are constant throughout the image, digital counts can easily be converted to radiance.

One can correct AVIRIS data for atmospheric effects (e.g., Green et al., 1993, Gao, et al., 1993). If the digital counts are corrected, we may then cast the problem in terms of apparent *reflectance* of the eth endmember seen by the ith spectral band, $R_{i,e}$

$$dc_i = gain_i \cdot reflectance_i + bias_i \tag{21}$$

$$reflectance_i = \sum_{e=1}^{n} R_{i,e} f_e \tag{22}$$

The spectral mixing can be done in terms of radiance (Equation 20) or, if corrected, in terms of reflectance (Equation 22). The choice of units depends upon the application. A reference library of endmembers would most likely be in units of reflectance. If the endmembers are derived from the image

data themselves, the units could be in radiance or digital counts. The equations that follow will be written in terms of reflectance, but the selection is arbitrary.

In summary, the spectral mixing equation can be written as

$$LRXS_i = \sum_{e=1}^{n} R_{i,e} f_e + \varepsilon_i \qquad (23)$$

where $LRXS_i$ are the ith band digital counts from the low resolution multispectral data converted to reflectance units, $R_{i,e}$ are the endmember reflectances from a stored library, f_e are the unknown endmember fractions, and ε_i are residual errors arising from errors in the endmember library, as well as unmodeled higher order effects.

Solving equation (23) results in n values of f_e at each pixel. The material maps are n different plots which map an f_e into a digital count (intensity) range. A greater fraction of an endmember makes the corresponding pixel brighter.

To implement (23) on an image of unknown endmembers, one searches a library of candidate endmembers (reflectance spectra) for the n endmembers that will minimize the residual error. Typically, a library of candidate endmembers is generated for a broad number of materials. There may be $L >> n$ potential endmembers in the library. Various sets of n endmembers are chosen, and fractions generated using equation (23). Almost all implementations of spectral mixing use the same endmembers for the entire scene. In this research, the algorithm is extended to unmix on a pixel by pixel basis.

Accurate spectral unmixing requires careful implementation (Adams et al., 1993). Unmixing spectral curves is an ill-posed problem (Price, 1994). There is no unique set of materials that combine to match the low resolution data. In color science, this same characteristic is known as metamerism. As a result, the entire unmixing process must be guided by significant user knowledge. Smith et al. (1990a) use an iterative approach, stopping when the number of endmembers is reasonable and the estimation residual is reduced to the level of image noise. A priori knowledge is essential to make the number of endmembers in the library reasonable.

Even with the best fitting set of n endmembers, residual error will remain. A shade endmember is used to account for shading and topographic effects (Smith et al., 1990a). This endmember is a

29

combination of real effects and an error term to account for some of the data spread. A shade endmember helps reduce the error and preserves the topographic information for plotting.

The dimensionality of remote sensing data is typically much less than the number of spectral bands in the sensor because of correlation in the data. However, the additional bands give increased confidence in the unmixed fractions (Sabol et al., 1992). It also allows the residual error to be used to deduce characteristics of unmodeled endmembers (Roberts et al., 1993).

Endmembers do not have to correspond to real objects. Smith et al. (1990a) describe a two-step process in which the digital counts are first related to endmembers, and then the endmembers are related to reference endmembers. The reference endmembers are real objects whose spectra are measured in a controlled environment. Aligning the endmembers allows for calibrating the digital counts in terms of reflectance spectra. Using an atmospheric correction algorithm should give good results without resorting to reference endmembers.

In some cases, the reflectance spectra do not combine linearly to generate the radiance at the sensor. For example, the relationship between leaf spectra and radiance is nonlinear due to the layered nature of tree canopies. Nonlinear spectral mixing models have been developed to relate leaf spectra and radiance (Roberts et al., 1990, Borel et al., 1991, and Borel and Gerstl, 1994). However, in many applications, one is only concerned with identifying the canopies, not the underlying leaves. In this situation, the nonlinear models are not needed since spectral mixing can be done with canopy, rather than leaf, endmembers. Intrinsic mixtures are not suitable for linear spectral unmixing.

If the scene contains materials for which the reflectance spectra are unknown, for example tree canopies, it may be possible to infer reflectance spectra from the data. Boardman et al. (1995) present a method of reducing the endmembers into those of interest and "background." Their methodology identifies the "purest pixels" by assuming that combinations of endmembers result in spectra that are numerically intermediate to the spectra of pure endmembers. This is the same assumption made in the spectral mixture analysis algorithm. If the pixel digital counts are mapped in an m-band spectral space, they naturally will form a convex region because of the mixing that occurs in many of the pixels.

Boardman et al. note that the extrema of this region correspond to pure pixels. The m-band coordinates of these pure pixels can be used as derived reflectance spectra for the endmembers.

In summary, spectral reflectances of real world objects exhibit tremendous variability. Temporal effects are especially evident in organic materials. Illumination and view angle are only partially compensated for by the shade endmember. The rest of the effects result in more variation from the library endmember spectra. This variation is manifested as errors in the material maps. Additionally, it is very difficult to quantify the results of a spectral unmixing algorithm since one requires a detailed ground truth.

To control the environment for the algorithm development reported here, SIG tools are used to completely define the images. Because the data are synthetically generated, the underlying unmixed solution is known. Additionally, perfectly unmixed images can be created to use in testing the sharpening process. This separates the unmixing and sharpening steps, and lets the effects of each be individually analyzed.

3.2 Sharpening

Given a set of low resolution fractions produced by the unmixing process described above, the next step is to spatially locate the fractions to the resolution of the sharpening band(s). The difficulty here is that there are many more unknowns than equations. Let s be the number of high resolution subpixels in a single hyperspectral superpixel, and constrain the subpixels to contain the same n materials as the superpixel. Then, there are s equations and $n*s$ unknowns. Sharpening is an underdetermined problem, requiring an optimization algorithm. Figure 11 shows a superpixel, its corresponding subpixels, and the unknown fractions for $n = 3$ endmembers and $s = 9$ subpixels.
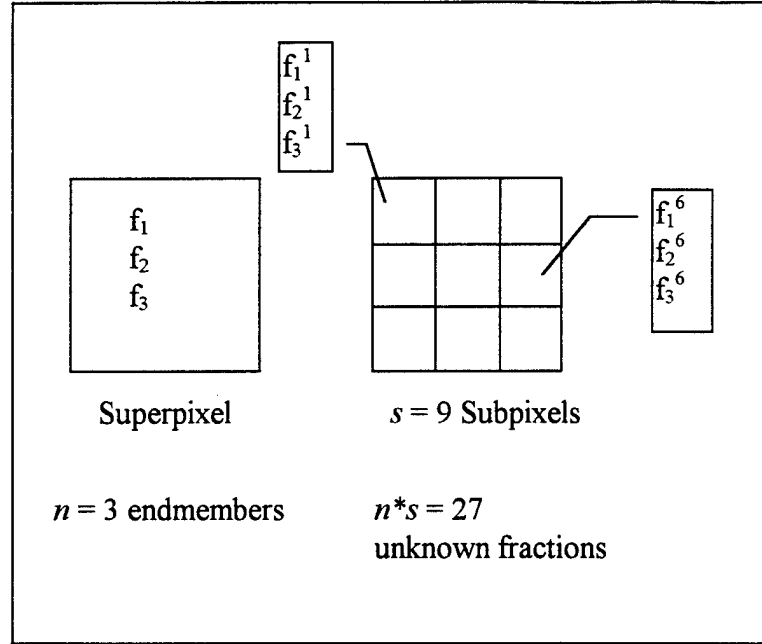
Figure 11: There Are Many High Resolution Unknowns in a Superpixel

The sharpening model takes the spectral unmixing form. The function to be minimized is the residual error in the superpixel digital counts (calibrated into reflectance units as in the spectral mixing algorithm),

$$HRP_j = \sum_{e=1}^{n} R'_{pan,e} f_e^{\,j} + \varepsilon_j; \quad j = 1 \ldots s \tag{24}$$

where, HRP_j is the jth element in of the High Resolution Panchromatic image corresponding to subpixel j, $R'_{pan,e}$ contains the library values for the reflectances of the appropriate endmembers in the sharpening band(s), and $f_e^{\,j}$ is the high resolution fraction for the eth endmember in the jth subpixel fraction vector.

We desire the values of $f_e^{\,j}$ that minimize equation (24) while maintaining consistency with the low resolution results. Consistency requires for each material, the high resolution fractions must average to the low resolution fraction,

$$\frac{1}{s}\sum_{j=1}^{s} f_e^{\,j} = f_e, \quad e = 1 \ldots n . \tag{25}$$

It is convenient to view this problem in "spreadsheet" form. For an example of $s = 4$ subpixels in a superpixel, and $n = 3$ endmembers from low resolution unmixing, the $n*s = 12$ unknowns are

32

represented in the center of the spreadsheet (Table 1). The consistency constraints require the rows sum to the appropriate values. To solve the optimization problem, the twelve fractions are adjusted to minimize the squared error in the digital counts, subject to the consistency constraints.

	Subpixel 1	Subpixel 2	Subpixel 3	Subpixel 4	Consistency Constraints
Endmember 1	f_1^1	f_1^2	f_1^3	f_1^4	$4*f_1$
Endmember 2	f_2^1	f_2^2	f_2^3	f_2^4	$4*f_2$
Endmember 3	f_3^1	f_3^2	f_3^3	f_3^4	$4*f_3$

Table 1: Spreadsheet Form of High Resolution Sharpening Problem

If all s high resolution digital counts are identical, the pixel is an aggregate mixture, and no sharpening is possible. If the panchromatic digital counts differ, the sharpening algorithm adjusts the high resolution fractions to minimize the residual error in (24).

3.3 Constraint Conditions

The preceding discussion presented spectral unmixing and sharpening as unconstrained problems. However, the underlying pixel represents a true mixture of one or more materials. Mixture problems are solved by constraining the coefficients.

The unmixing literature contains examples using each of the constraint strategies. Some investigators use unconstrained unmixing and screen out unrealistic solutions. Others use partial constraints by requiring the fractions to sum to unity. The fully constrained approach is used less often. Applying inequality constraints makes the algorithm more complex. The fully constrained case is also likely to be susceptible to spectral variation between different material samples and between the samples and the spectral library.

Define three different constraint conditions. The first is *unconstrained*, where the fractions can take on any value needed to minimize the residual error. However, the unconstrained sharpening problem is defined to include the consistency constraints in equation (25).

33

The second condition is called *partially constrained*. Here, the fractions within a pixel are required to sum to unity. The sum is taken over all the materials in the appropriate pixel. At low resolution there is one constraint

$$\sum_{e=1}^{n} f_e = 1 \tag{26}$$

while at high resolution, this provides s equality constraints.

$$\sum_{e=1}^{n} f_e^j = 1, \quad j = 1...s \tag{27}$$

However, only $(s - 1)$ of the constraints are independent. If equality constraints were included in the Table 1 spreadsheet, they would require each column sum to one.

A *fully constrained* set would also require each individual fraction to lie between zero and one. There are $2*n$ inequality constraints at low resolution

$$0 \le f_e \le 1, \tag{28}$$

and $2*n*s$ inequality constraints at high resolution

$$0 \le f_e^j \le 1, \quad j = 1...s. \tag{29}$$

The inequality constraints are not all independent, but an optimization algorithm only applies the *active* inequality constraints during any search. Problems containing inequality constraints, such as the fully constrained problem, require an iterative solution.

Although fully constrained fractions seem intuitive for a mixture problem, values outside the bounds have some physical meaning. A two dimensional example is useful. Figure 12 illustrates mixtures of three materials in two spectral bands. Reflectance values in each of two spectral bands form the coordinate system. The 'X' symbols indicate the three pure materials. For example, one material has reflectance [0.5, 0.8]. The 'O' symbols represent three different mixtures: a pure material, an equal mixture of all three materials, and a 50% mixture of two materials. The pure material has a spectral response of an endmember, while the equal mixture response lies at the center (average) of the endmembers. The triangle formed by connecting the X's describes the region of all possible mixtures of

34

these materials. In other words, given the three endmembers defined by 'X,' all mixtures would plot within the triangle.
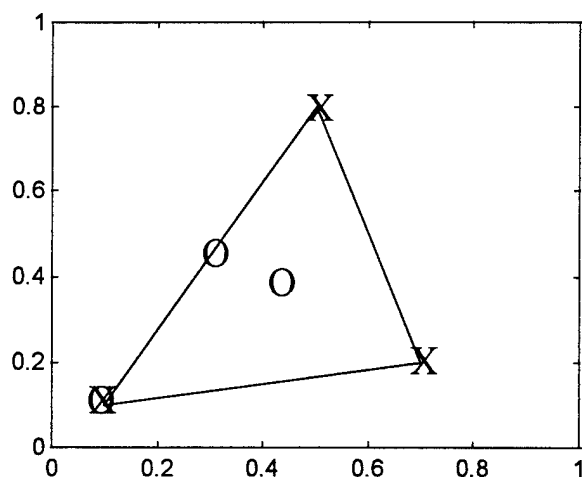
Figure 12: Three Material Mixtures in Two Spectral Bands

Figure 13 illustrates the effect of random variation on the endmember location. Endmembers are derived from a library or from pure pixels within the image. To reduce noise, averages of many samples are used. Therefore, although the endmembers are plotted as single points, they really represent *mean* endmember vectors. If the real materials exhibited reflectance spectra that were gaussian distributed about the mean, their distribution could be described with a covariance matrix. The three endmembers from Figure 12 are plotted as the vertices of the dashed triangle. Here, the triangle delineates the set of expected mixtures. Around each of these endmembers are drawn two contour curves, representing equally likely departures from the mean values. Suppose *in a particular pixel*, the three materials had reflectance values represented by the vertices of the solid triangle. Then, the actual set of possible mixtures is contained within the solid triangle.
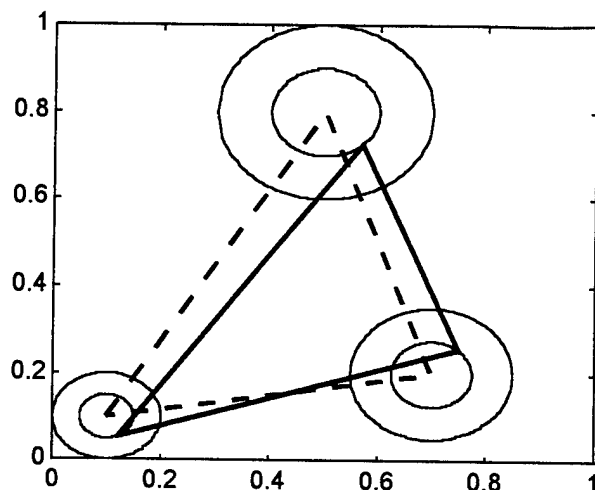
Figure 13: Gaussian Distributed Endmembers

The example is completed by examining a hypothetical reflectance measurement. Suppose the measured reflectance values for this pixel were [0.62, 0.5]. This point is plotted as an 'X' in Figure 14. The triangles from the previous figure are repeated for clarity. In the "true" space relative to the solid triangle, the fractions are [0.05, 0.54, 0.41]. These fractions satisfy all requirements for a fully constrained mixture. However, in terms of the "average" endmembers represented by the dashed triangle, the fractions are [-0.03, 0.49, 0.54].
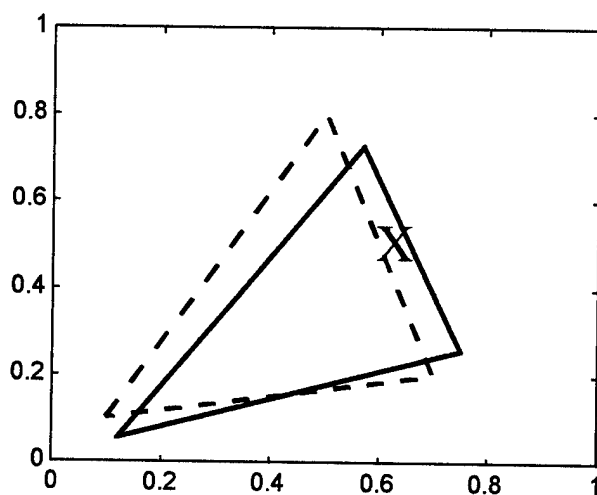
Figure 14: Mixture Requiring Negative Fractions

As this example shows, if the library reflectance values are derived from an "average" material response, fractions greater than one and deviations in the "negative" direction from other endmembers are expected. An algorithm may need to recognize small negative fractions and fractions slightly greater than one as acceptable, while preventing solutions with unreasonable fractions.

This section presented a conceptual overview of the fusion algorithm. The next section provides the mathematical framework for solving the spectral unmixing and sharpening problems under the different constraint conditions.

4. Algorithm Development

The numerical techniques required by the unmixing and sharpening algorithms require solving linear least squares (LS) problems. LS problems are addressed in many mathematics and engineering disciplines. The notation is nonstandard, and there are often multiple solution techniques. This section examines the mathematical foundation necessary to solve linear LS problems, and forms the framework for the unmixing and sharpening algorithms. Section 4.1 contains many subsections, and steps through the various LS solution techniques. This section is the "tool set" one must use to understand the algorithms. Section 4.2 uses the tools to solve the spectral unmixing problem. Section 4.3 applies the tools to the sharpening algorithm.

4.1 Optimization and the Least Squares Problem

This theory section will review the "tool set" required to solve the various LS problems that arise in this image fusion algorithm. The fusion algorithm will attempt to "optimize" the error in the digital counts in a minimum (least) squared error sense. The first subsection reviews the necessary and sufficient conditions for minimizing a function. The second subsection uses regression techniques to solve the general LS problem. Over-determined and under-determined problems are described, and the basic LS solution is explained.

The third subsection presents a tool for analyzing LS solutions, the Analysis of Variance (ANOVA). Section 4.1.4 applies the ANOVA technique to the problem of subset selection. This idea improves upon traditional unmixing by allowing different endmembers to be used on a pixel by pixel basis. Removing unnecessary endmembers from the model increases the accuracy of the remaining fractions.

Finally, the fifth subsection is a lengthy analysis of the effect of constraints on the LS problem solution. Equality and inequality constraints are presented. The problems are solved with Lagrange multipliers and orthogonal decomposition techniques. Although Lagrange multipliers give an intuitive insight into the effect of the constraints, the decomposition technique is a preferred implementation for

digital computers. Many software packages have decomposition routines with excellent numerical stability.

At the completion of this section, the tools described here are related to solving the specific unmixing and sharpening problems.

4.1.1 Necessary Conditions

Consider a general function $F(u)$, and let u^* be the value of u at which $F(u)$ is minimized. Also define

$$\Delta F = F(u) - F(u^*). \tag{30}$$

A Taylor series expansion of $F(u)$ about u^* is

$$F(u) = F(u^*) + \frac{\partial F}{\partial u}\Big|_{u^*} \cdot \Delta u + \frac{1}{2} \frac{\partial^2 F}{\partial u^2}\Big|_{u^*} \cdot \Delta u^2 + \cdots. \tag{31}$$

Examine a first order expansion,

$$\Delta F = \frac{\partial F}{\partial u}\Big|_{u^*} \cdot \Delta u. \tag{32}$$

If $F(u^*)$ is the minimum, $\Delta F > 0$. Therefore, since Δu is arbitrary, a necessary condition for a minimum is that the first derivative must equal zero.

$$\frac{\partial F}{\partial u}\Big|_{u^*} = 0 \tag{33}$$

Points which satisfy (33) are called *stationary* points. Now examine the second order term

$$\Delta F \approx \frac{\partial^2 F}{\partial u^2}\Big|_{u^*} \cdot \Delta u^2. \tag{34}$$

While we still must have $\Delta F > 0$, $\Delta u^2 > 0$ for any deviation from u^*. Therefore, a sufficient condition for a minimum is

$$\frac{\partial^2 F}{\partial u^2}\Big|_{u^*} > 0. \tag{35}$$

If $\partial^2 F/\partial u^2 = 0$, u^* is a *singular* point, and one would check higher order derivatives to confirm the function is minimized. A necessary condition for a minimum is

39

$$\left.\frac{\partial^2 F}{\partial u^2}\right|_{u^*} \geq 0.$$ (36)

If u is an m-vector (a vector with m elements), $\partial^2 F/\partial u^2$ is an m x m matrix. Alternatively, the necessary condition implies this matrix must be positive semidefinite, which means its eigenvalues must be zero or positive. If the matrix has only positive eigenvalues, it is positive definite, and the sufficient condition is satisfied (Bryson and Ho, 1969).

Algorithms that search (iterate) for a solution often use the first derivative as the primary mechanism for finding the minimum.

4.1.2 The General LS Problem

In this section, the fundamental LS problem is described. Consider a predictive model

$$y = \xi_0 + \sum_{i=1}^{n} A_i \xi_i + \varepsilon,$$ (37)

where y is linearly related to n predictor variables $A_1, A_2, ..., A_n$ through unknown coefficients $\xi_0, \xi_1, ..., \xi_n$. Assume specific realizations of this model are corrupted by random deviations, ε, which are independently sampled from a distribution with zero mean and variance σ^2. In matrix form, the process model is

$$y = A\xi + \varepsilon,$$ (38)

where $A_0 = 1$ allows a constant coefficient ξ_0 to enter the model. We desire to estimate the unknown coefficients ξ, with estimates x, which then are used to predict y

$$\hat{y} = Ax.$$ (39)

If there are m realizations, or measurements, of the underlying process, y is an m-vector, A is an m x n matrix, and x is an n-vector. This can be represented as a system of equations

$$y_1 = A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n$$
$$y_2 = A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n$$
$$\vdots$$
$$y_m = A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n$$

(40)

The unknown coefficients can be estimated via a least squares (LS) procedure.

The LS estimate (x) has certain properties. It gives the minimum squared error, regardless of the distribution of ε. The estimate is the minimum variance, unbiased, linear estimator of ξ. Finally, if the residuals are normally distributed, x is also the maximum likelihood estimate (Draper and Smith, 1981).

The goal is to minimize $(y - \hat{y})^2$, which in matrix algebra terms is written as minimizing the norm (euclidean length) of the vector difference. The general LS problem may be stated (Lawson and Hanson, 1974):

Given a real m x n *matrix* A *of rank* k $\leq$ *min* (m,n), *and given a real* m-vector y, *find a real* n-vector x_0 *minimizing the euclidean length of* y - Ax.

$$\min \| y - \hat{y} \| = \min \| y - Ax \|$$

(41)

The solution techniques vary depending upon the relative dimensions of the variables, as well as the rank of A. In the most typical case, there are more measurements than unknowns $(m > n)$, and the predictor variables are independent (rank $(A) = n$). Figure 15 illustrates the different possibilities for the LS problem, with case 2 corresponding to the most typical circumstances.
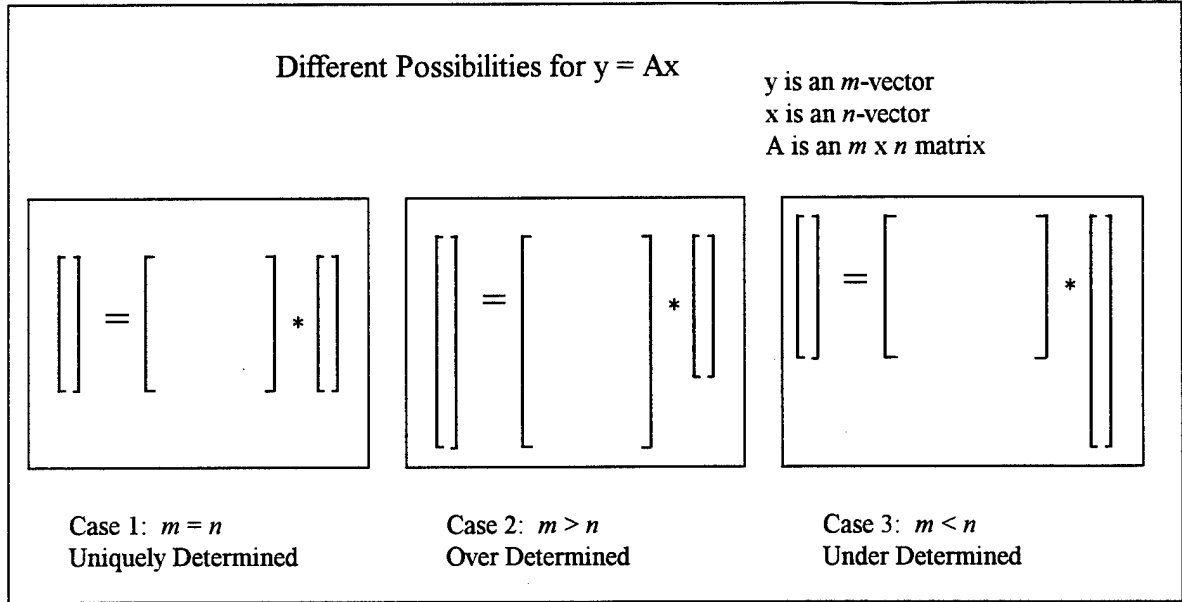
Figure 15: Three Types of Least Squares Problems

If $m = n$, and A is full rank (Case 1), there is a unique solution. One inverts A directly to solve

$$x = A^{-1}y. \tag{42}$$

A more general notation which applies to all cases is

$$x = A^{\#}y \tag{43}$$

where $A^{\#}$ is called the pseudoinverse.

Case 2 is a classical regression problem. One forms the "normal" equations by premultiplying (39) by A',

$$A'y = A'Ax. \tag{44}$$

Note the normal equations can also be obtained by minimizing the quadratic function.

$$
\begin{aligned}
F(x) &= (y - \hat{y})^2 \\
&= (y - Ax)^2 \\
&= y'y - 2y'Ax + x'A'Ax
\end{aligned} \tag{45}
$$

The minimum occurs at the stationary points, where the first derivative is zero,

$$0 = \frac{\partial F}{\partial x}$$
$$0 = -2A'y + 2A'Ax \tag{46}$$
$$A'y = A'Ax$$

Since A' is $n \times m$, this forms n equations which can be solved for the n unknowns in x. The conditions for directly solving this are that $A'A$ is invertible. If rank $(A) = n$ (all n predictors are independent), we can invert $A'A$ to solve for x

$$x = (A'A)^{-1}A'y \tag{47}$$

For the over-determined case (Case 2), where $m > n$ and rank $(A) = n$, the pseudoinverse is defined as in (47)

$$A^{\#} = (A'A)^{-1}A'. \tag{48}$$

In the under-determined case (Case 3), where m < n and rank $(A) = m$, the pseudoinverse is

$$A^{\#} = A'(AA')^{-1}. \tag{49}$$

These equations for the pseudoinverse give a quick solution to the full rank LS problem. However, as the following section will show, further analyzing the LS solution provides useful information. This is especially true for the over-determined case corresponding to the typical LS regression problem (Draper and Smith, 1981). One can calculate the fitted values

$$\hat{y} = Ax, \tag{50}$$

and the vector of *residuals*

$$e = y - \hat{y}, \tag{51}$$

which has the properties

$$\sum_{i=1}^{m} e_i \hat{y}_i = 0, \tag{52}$$

and, if x_0 is included in the estimate,

43

$$\sum_{i=1}^{m} e_i = 0. \tag{53}$$

The coefficients have a covariance

$$V(x) = (A'A)^{-1}\sigma^2. \tag{54}$$

Define A_0 as a 1 x n vector whose elements are in the same form as a row of A. Then,

$$\hat{y}_0 = A_0 x \tag{55}$$

is the fitted value at a specified point A_0. In other words, $\hat{y}_0$ is the value predicted at A_0 by the regression equation. If A_0 corresponds to the reflectance contribution of several materials at a particular wavelength, $\hat{y}_0$ would be the predicted combined reflectance at that wavelength. It would have variance of

$$V(\hat{y}_0) = A_0'(A'A)^{-1} A_0 \sigma^2. \tag{56}$$

Finally, an analysis of variance (ANOVA) can be performed to study how much of the total variance is explained by the regression model, and how much is due to the random residuals. The following sections use the ANOVA formulation to examine the fit of regression models. This concept can then be used to unmix each individual pixel in the image.

4.1.3 Analysis of Variance

An ANOVA calculation is derived from the following basic identity (Draper and Smith, 1981),

$$e_i = y_i - \hat{y}_i = (y_i - \bar{y}) - (\hat{y}_i - \bar{y}). \tag{57}$$

This says the residual is the difference between two quantities: (a) the deviation of the observed y_i from the overall mean, and (b) the deviation of the fitted $\hat{y}_i$ from the overall mean. Figure 16 (Draper and Smith, 1981) is a geometrical interpretation of this identity.

44
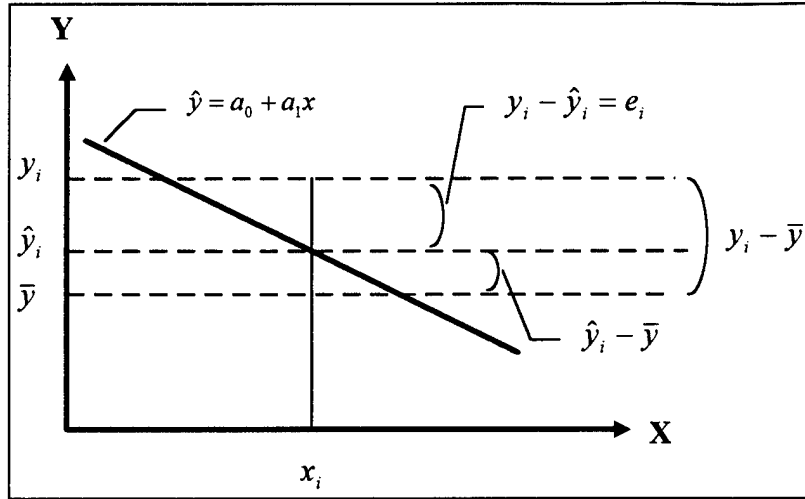
Figure 16: Geometrical Interpretation of Residual

Rearranging (57), squaring both sides, and summing over all m measurements gives

$$\sum (y_i - \bar{y})^2 = \sum (\hat{y}_i - \bar{y})^2 + \sum (y_i - \hat{y}_i)^2 ,$$ (58)

where the cross product term is shown to be zero by using the following relationship

$$\hat{y}_i = \bar{y} + x\left(A_i - \bar{A}\right),$$ (59)

and

$$
\begin{aligned}
2\sum (\hat{y}_i - \bar{y})(y_i - \hat{y}_i) &= 2\sum \left\{\bar{y} + x\left(A_i - \bar{A}\right) - \bar{y}\right\}\left\{y_i - \bar{y} - x\left(A_i - \bar{A}\right)\right\} \\
&= 2\sum \left\{x\left(A_i - \bar{A}\right)\right\}\left\{y_i - \bar{y} - x\left(A_i - \bar{A}\right)\right\} \\
&= 2\left\{x\left(m\bar{A} - m\bar{A}\right)\right\}\left\{m\bar{y} - m\bar{y} - x\left(m\bar{A} - m\bar{A}\right)\right\} \\
&= 0
\end{aligned}
$$ (60)

Without making distributional assumptions on the residuals, the total variance can be divided as follows: The term on the left hand side of (58) is the total variance, called the Sum of Squares (SS) *about the mean*. The first term on the right hand side is the SS *due to the regression* model. The second term is the SS *about the regression*, or the SS due to the residual.

In an ANOVA analysis, a table is formed tracing how the variance can be explained (Table 2). The first column shows the variation source. The second column indicates the degrees of freedom (dof) associated with the measurement. The third column shows how the Sum of Squares would be calculated

45

for that source. The SS are annotated as 'corrected' because the mean of y is subtracted from the measurements. The fourth column indicates the dof corresponding to the uncorrected measurements. The fifth column shows how to calculate the SS in matrix algebra form. The final column, mean square (MS), is calculated by dividing the SS by the corresponding degrees of freedom. The MS due to the residual is an estimate of the residual variance, σ^2.

Source	dof (corrected)	SS	dof (uncorrected)	SS (matrix form)	MS
Due to Regression	n-1	$\sum(\hat{y}_i - \bar{y})^2$ (corrected)	n	x'A'y	MS(regression)
About Regression (residual)	m-n	$\sum(y_i - \hat{y}_i)^2$	m-n	y'y - x'A'y	MS(residual) $\approx$ s^2
Total	m-1	$\sum(y_i - \bar{y})^2$ (corrected)	m	y'y (uncorrected)	

Table 2: Basic ANOVA Table

Since y is an m-vector, there are m degrees of freedom in the total uncorrected SS. Since the regression model contains n terms, there are n degrees of freedom in the uncorrected SS due to regression. In an ANOVA table, the Sum of Squares and the degrees of freedom add. Therefore, SS and dof due to the residual are found by subtracting the amounts due to regression from the totals.

One can subdivide the variation sources. For example, the uncorrected SS(regression) and SS(total) can be expressed as the sum of variation explained by a constant term (x_0) with a single dof, and the remaining (corrected) variation with (n-1) dof. This corrects for the mean of the measurements. If one designs an experiment to have multiple measurements at a single condition, the SS(residual) can be divided into terms showing lack of fit and pure error. For the image fusion algorithm, these modifications are unnecessary.

The benefit of the ANOVA table becomes apparent when one also makes distributional assumptions about the errors. If $\varepsilon \sim N(0, I\sigma^2)$ (gaussian, zero mean, and independent), then the sum of m normalized squared errors has a chi-square distribution with m dof, written χ^2_m (Dobson, 1990). If the regression model is a good one, then the errors should be chi-square distributed. If the model is poor, the errors will not be chi-square. Thus, a hypothesis test can examine the errors, and determine, to a specified confidence level, whether the proposed model is adequate. The hypothesis test uses the relationship that

46

the ratio of two chi-square variables divided by their respective degrees of freedom has an F-distribution with the corresponding degrees of freedom. This is denoted

$$\frac{\chi_m^2/m}{\chi_n^2/n} \sim F_{m,n}. \tag{61}$$

The overall regression equation is tested by defining a null hypothesis that all the coefficient terms should be zero. The alternate hypothesis is that at least one of the terms is nonzero.

$$
\begin{aligned}
H_0: &\quad \xi_1 = \xi_2 = \cdots = \xi_n = 0 \\
H_1: &\quad \text{not all } \xi_i = 0
\end{aligned} \tag{62}
$$

Consider the distribution of two specific random variables, SS(regression) and SS(residual). If the errors are gaussian, the sum of squares variables are χ^2 distributed. They are normalized by their respective degrees of freedom to create the mean square variables.

$$\frac{SS(regression)}{n-1} = MS(regression) \sim \frac{\chi_{n-1}^2}{n-1} \tag{63}$$

$$\frac{SS(residual)}{m-n} = MS(residual) \sim \frac{\chi_{m-n}^2}{m-n} \tag{64}$$

The ratio

$$F = \frac{MS(regression)}{MS(residual)} \tag{65}$$

follows an $F_{n-1,m-n}$ distribution as long as $\xi_i = 0$.

One forms the ratio of the MS(regression) to the MS(residual), and compares this to a tabulated F-statistic with n-1, and m-n degrees of freedom, and a desired confidence level. If the ratio is greater than the tabulated value, one *rejects* the null hypothesis in favor of the regression model. Large ratios occur when the regression model explains a large portion of the variance. If the ratio was smaller than the F statistic, one *would not reject* the null hypothesis. The variation is such that the null hypothesis may be feasible. In this case, the model does not explain enough of the variance to warrant using it. Relative to the proposed model, the null hypothesis may be valid. A better model is required.

47

In the LS problems associated with unmixing and sharpening, it is of limited value to test the entire regression model. This is a fundamental assumption behind spectral mixture analysis. The real benefit of an ANOVA table and an F-test comes from investigating the value of adding terms to the model.

4.1.4 Subset Selection

Traditional unmixing fixes the number of endmembers to use throughout the entire image. The following section describes an approach which can be used to select the appropriate endmembers on a pixel by pixel basis. By only including a subset of the endmember library in the regression model, a more robust prediction results. When excess endmembers are included (e.g., to map more material classes), the extra terms cause overfit in the model. Conversely, if one uses just a few endmembers, the fractions are improved, but fewer materials can be mapped.

To unmix each pixel, one must identify which n predictors (endmembers) are the proper terms for the model. Consider the case where one has a library of L potential endmembers, but where a priori knowledge indicates the actual number of terms in the model should be $n \ll L$. In other words, we may have a library of materials which we expect to find in the image, but not all of them will be found in any one pixel. The problem is made more difficult because n is unknown. One must determine the proper subset of endmembers to use (Miller, 1990).

Miller warns of the dangers inherent in subset selection. Adding terms to a prediction model *increases* the variance of the prediction, e.g. equation (56), with the benefit of *decreasing* the prediction bias. A constant model has zero variance, but a large prediction bias. Consider the case where the true variables, ξ, are divided into a set which are entered into the prediction model, ξ_A, and a set which are not included in the model, ξ_B. Then, the coefficients are found via (47)

$$x_A = \left(A_A{}' A_A \right)^{-1} A_A{}' y \qquad (66)$$

and the expected value of x_A is

$$E(x_A) = (A_A' A_A)^{-1} A_A' A\xi$$
$$= (A_A' A_A)^{-1} A_A' [A_A \quad A_B]\xi$$
$$= (A_A' A_A)^{-1} [A_A' A_A \quad A_A' A_B]\xi \tag{67}$$
$$= \xi_A + (A_A' A_A)^{-1} A_A' A_B \xi_B$$

The second term in (67) is the bias in the first set of regression coefficients (x_A) arising from the omission of the last set of variables (Miller, 1990). The goal is to add variables to reduce the omission bias without excessively increasing the variance of the estimate.

Miller also warns about selection bias. The LS development assumes the variables have been chosen independently of the data being used to generate the regression coefficients. One should separate the variable selection process from the coefficient estimation step. Since this is rarely done in practice, regression models can suffer from selection bias. The accuracy of the following analysis is subject to omission and selection biases. Nevertheless, by carefully designing the algorithm, and conservatively interpreting the results, the effects of these biases can be mitigated.

One approach to finding the 'best' subset is to examine *all possible subsets*. For any given value of n, the number of combinations of L choose n is $L!/n!(L-n)!$. For libraries of any substantial size, or for models with any more than a few terms, the computational burden becomes prohibitive. An additional complication is that adding terms to the model always reduces the residual error (i.e., less omission bias). After a point, additional terms will fit the random error instead of the underlying process. This condition is known as overfit, and results in overly complex models with poor predictive properties.

A procedure called *forward regression* incrementally adds terms to the model, stopping when new terms are likely to be overfit to the errors. Another technique called *backward regression* starts with all L terms in the model, and removes those with poor predictive properties. An algorithm called *stepwise regression* combines the characteristics of both forward and backward regression. At each stage of the algorithm, the best variable is added, and the already accepted terms are examined for removal.

Stepwise regression is based upon an ANOVA calculation of the "Extra Sum of Squares," (Draper and Smith, 1981). In this analysis, one compares an n-term model with an $(n-1)$-term model to calculate the benefit of adding the additional term. Define the reduced term model by

$$\hat{y} = Wz; \quad z = (W'W)^{-1}W'y \tag{68}$$

where z is an $(n-1)$-vector and W is an $m \times (n-1)$ matrix. The SS(regression) with n dof can be divided into the SS due to the reduced order model with $(n-1)$ dof and the SS due to the extra term in the model with a single dof. The ANOVA structure is shown in Table 3.

Source	dof (uncorrected)	SS	MS
SS reduced model	n-1	z'W'y	
SS extra term	1	x'A'y - z'W'y	MS(extra term)
Due to Regression	n	x'A'y	
About Regression (residual)	m-n	y'y - x'A'y	MS(residual) $\approx$ s^2
Total	m	y'y (uncorrected)	

Table 3:   Extra Sum of Squares ANOVA Table

As with the general ANOVA, the sum of squares variables are χ^2 distributed. A ratio of the MS(extra term) to the MS(residual) can be compared to an F-statistic with 1 and $m-n$ degrees of freedom, at a specified confidence level. For this test, the null hypothesis is that the coefficient corresponding to the extra term is zero (unnecessary). If the MS ratio is greater than the F-statistic, we reject the null hypothesis and conclude the more complex model is required. If the MS ratio is smaller than the F-statistic, we would not reject the simpler model, which does not include the extra term.

Stepwise regression uses this sequential F-test to add and remove terms to the regression model. Note it is not required for the ratios to actually have an F distribution. In fact, because the ratios formed from wrong models are *not* F distributed, an F-table is rarely used. More often, a fixed value of F-to-enter and F-to-remove is used regardless of the specific degrees of freedom in a particular model under test (Kennedy and Gentle, 1980).

A spectral unmixing algorithm based upon stepwise regression calculates the proper number of endmembers to include on a pixel by pixel basis. This algorithm maps a greater number of materials while preventing the fractions from being overfit to the image noise.

4.1.5 Handling Constraints

The above description applies to solving unconstrained linear least squares problems. With a mixture problem, however, one knows the fractions must sum to unity. Additionally, negative fractions and fractions greater than one cannot occur. If the coefficients in a model are constrained, one must solve a *restricted least squares* problem. In particular, we are interested in linear, equality and inequality constraints.

4.1.5.1 Equality Constraints

Equality constraints reduce the dimensionality of the solution space. They could be used to explicitly eliminate variables in the objective function, leaving a smaller number of unconstrained variables. Alternatively, a least squares problem restricted by linear equality constraints can be solved via Lagrange multipliers or via linear algebra. Both derivations are presented.

Suppose we desire to minimize a function F(u), subject to p equality constraints

$$c_i(u) = d_i; \quad i = 1...p.$$ (69)

One can form an augmented function called the Lagrangian by adding terms proportional to the constraints,

$$L(u) = F(u) + \sum_{i=1}^{p} \lambda_i \left(c_i(u) - d_i \right),$$ (70)

where the λ_i are called Lagrange multipliers. Since we require $c_i(u) = d_i$ at the solution point, the minimum of $L(u)$ will be the same as the minimum of $F(u)$. Applying the first order necessary condition to L gives

$$\frac{\partial L}{\partial u} = \frac{\partial}{\partial u} F(u) + \sum_{i=1}^{p} \lambda_i \frac{\partial}{\partial u} c_i(u) = 0 \qquad (71)$$

$$\frac{\partial L}{\partial \lambda_i} = c_i(u) - d_i = 0; \quad i = 1...p. \qquad (72)$$

For the LS problem with linear constraints, (69) becomes

$$Cx = d, \qquad (73)$$

where C is a $p \times n$ matrix, x is an n-vector, and d is a p-vector. For the LS problem, the function to be minimized is the squared error

$$\min F(x) = (y - \hat{y})^2 = (y - Ax)^2 = y'y - 2y'Ax + x'A'Ax$$
$$\text{subject to } Cx = d \qquad (74)$$

The Lagrangian is

$$L(x) = F(x) + \lambda'(Cx - d), \qquad (75)$$

where the multipliers are written as a p-vector. The first derivative necessary conditions are

$$\frac{\partial L}{\partial x} = -2A'y + 2A'Ax + C'\lambda = 0 \quad \text{(n equations)}$$
$$\frac{\partial L}{\partial \lambda} = Cx - d = 0 \quad \text{(p equations)} \qquad (76)$$

Let the multiplier account for the factor of two in the first equation. In matrix form, (76) can be written as

$$\begin{bmatrix} A'A & C' \\ C & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} A'y \\ d \end{bmatrix} \qquad (77)$$

If A'A is nonsingular, the above matrix can be inverted to solve for x and λ. Since these calculations are performed in a computer, the augmented system (77) can be solved numerically using standard algorithms. The first n elements of the solution vector correspond to the coefficients, while the last p elements are the multipliers. An analytical solution (Appendix A) is

$$x = x_u + (A'A)^{-1}C'\left[C(A'A)^{-1}C'\right]^{-1}(d - Cx_u) \qquad (78)$$

52

where $x_u = (A'A)^{-1}A'y$ is the usual unrestricted estimator (Draper and Smith, 1981).

Lawson and Hanson (1974) present a linear algebra solution to the constrained LS problem using an orthogonal coordinate transformation. Once the constrained variables are accounted for, a reduced dimension unconstrained LS solution finds the remaining unknowns.

We desire to minimize $\| y - Ax \|$, subject to $Cx = d$. Assuming the constraints are consistent, rank(C) $= p < n$. An orthogonal decomposition can be used to partition C into an $p \times p$ submatrix C_1 and a $p \times (n\text{-}p)$ zero matrix,

$$CV = C\left[V_1 \mid V_2\right] = \left[C_1 \mid 0\right], \tag{79}$$

where V is an $n \times n$ orthogonal matrix that when postmultiplied times C, partitions C as desired. Therefore, V_1 is $n \times p$, and V_2 is $n \times (n\text{-}p)$. A V matrix can be found by a variety of algorithms. For example, a singular value decomposition of C

$$\begin{aligned} C &= USV' \\ S &= U'CV \end{aligned} \tag{80}$$

will suffice. In the singular value decomposition, U is a $p \times p$ orthogonal matrix, V is a $n \times n$ orthogonal matrix, and S is a $p \times n$ matrix of the singular values. A generalized pseudoinverse can also be defined using the singular value decomposition.

One can use the decomposition provided by V to solve a p-dimensional subsystem

$$\begin{aligned} C_1 w_1 &= d \\ w_1 &= C_1^{-1}d \end{aligned} \tag{81}$$

for the p-vector w_1, where C_1 is invertible because of the decomposition.

Similarly, one uses V to decompose the LS system into the same coordinates.

$$AV = A\left[V_1 \mid V_2\right] = \left[A_1 \mid A_2\right] \tag{82}$$

where A_1 is $m \times p$, and A_2 is $m \times (n\text{-}p)$. To solve the LS problem, first subtract the effect of the transformed constraints from the measurements

$$\tilde{y} = y - A_1 w_1. \tag{83}$$

Then solve a smaller dimension LS problem for the remaining $(n\text{-}p)$ variables, w_2

$$\widetilde{y} = A_2 w_2$$
$$w_2 = \left(A_2{'}A_2\right)^{-1}A_2{'}\widetilde{y}$$

(84)

Finally, the intermediate results are transformed back to the original coordinate system by

$$x = Vw = \left[V_1 \mid V_2\right]\begin{bmatrix} w_1 \\ \hline w_2 \end{bmatrix}.$$

(85)

The Lagrange multiplier and the orthogonal decomposition techniques yield identical results. Selection of one method over the other is a personal preference.

4.1.5.2 Inequality Constraints

Inequality constraints are fundamentally different than equality constraints. Whereas each equality constraint reduces the number of free variables in the LS problem, inequality constraints establish boundaries within the solution space (Simmons, 1975). At the optimum point, *inactive* inequality constraints have no effect. A scheme is needed to iterate through the solution space, determining which are the active constraints. Once the *active* constraints are known, they can be treated (locally) as equality constraints to solve a restricted LS minimization. An algorithm then searches for another solution point, determines the active constraints, and improves upon the minimization. Since solving inequality constrained LS problems requires iterating, run time for the fully constrained algorithm is longer.

If a constrained local minimum occurs at a boundary, the first partial derivatives may not equal zero. Thus the first order necessary conditions (33) are an insufficient test for a minimum. A more general set of conditions are needed. The *Kuhn-Tucker conditions* are the generalized first order necessary conditions used when a problem includes inequality constraints.

Consider minimizing F(u) subject to r inequality constraints of the form

$$g_i(u) \le h_i; \quad i = 1\ldots r.$$

(86)

Note that a choice of sign convention is required. Any constraints $g_i(u) \ge h_i$ must be multiplied through by -1 to conform to the stated convention. Also, the h_i term could easily be removed by placing it into the

54

$g_i(u)$ function. As before, form the Lagrangian by augmenting the objective function with each constraint multiplied by a Lagrange multiplier. To minimize F, the following Kuhn-Tucker conditions must be satisfied

$$\frac{\partial}{\partial u} F(u) + \sum_{i=1}^{r} \lambda_i \frac{\partial}{\partial u} g_i(u) = 0 \tag{87}$$

$$g_i(u) \leq h_i; \quad i = 1...r \tag{88}$$

$$\lambda_i \geq 0; \quad i = 1...r \tag{89}$$

$$\lambda_i \left[h_i - g_i(u) \right] = 0; \quad i = 1...r \tag{90}$$

The first two equations are familiar. One is the first derivative of the Lagrangian with respect to the minimization variable. The second is the original inequality constraints. Kuhn-Tucker generalizes the Lagrange multipliers by adding the last two restrictions. In the third equation, we see the multipliers must be zero or positive for the solution to be a minimum. If λ_i were negative, this means $\partial L/\partial g_i < 0$, and the objective function could be reduced.

The last restriction is the key. In order for (90) to be satisfied, either a constraint must be an equality (active), or its multiplier must be zero. If a particular multiplier were positive while its constraint was inactive, it would be possible to reduce the cost function by moving towards the boundary in question. Therefore, at the minimum, all inactive constraints have a zero Lagrange multiplier, and the multiplier for all active constraints is positive (Simmons, 1975).

A graphical interpretation of the constraints is helpful in visualizing their effect. Equation (87) requires the gradient of F be expressible as a linear combination of the gradients of the constraints. ∇F must be parallel, and in the opposite direction, to the linear combination of ∇g_i. In words, "the gradient of F with respect to u at a minimum must be pointed in such a way that decrease of F can only come by violating the constraints," (Bryson and Ho, 1969). Figure 17 (Bryson and Ho, 1969) shows a cost function with a single constraint. Since ∇g is not parallel to ∇F, one can move into the shaded region and reduce F.
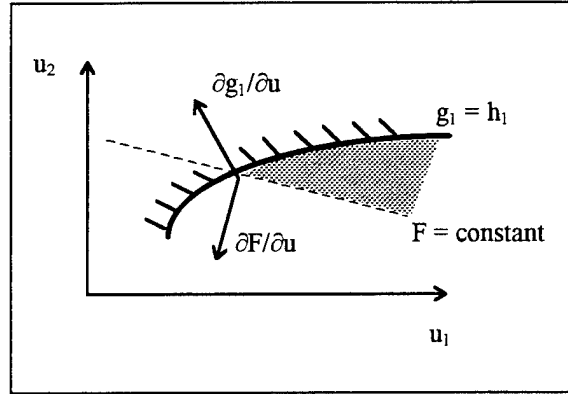
Figure 17: If ∇G is Not Parallel to ∇F, the Function is Not Minimized

In Figure 18 (Bryson and Ho, 1969), two constraints illustrate that at the minimum, ∇F can be formed as a (negative) linear combination of the ∇g_i.
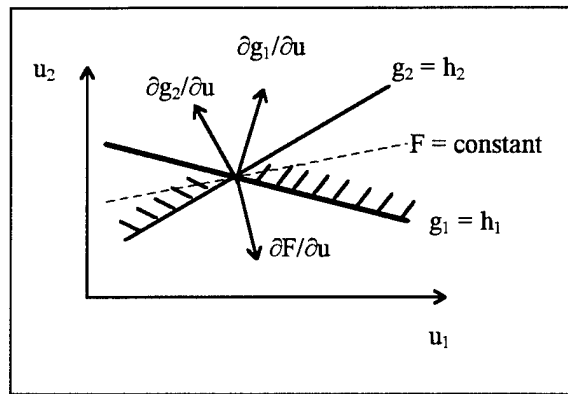
Figure 18: At a Minimum, ∇F is a Linear Combination of ∇g$_i$

The Kuhn-Tucker conditions provide the basis for algorithms to find the minimum. As the algorithms iterate through the solution space, the Kuhn-Tucker conditions are checked to determine if a constrained minimum is reached.

At this point, write the constraints as a linear function of x to simplify the notation. In the general case, one would linearize the constraints at each stage of the iteration. Equation (86) becomes

$$Gx \leq h,\tag{91}$$

where G is an $r \times n$ matrix, and h is an r-vector.

Define the Least Square Inequality (LSI) problem as:

56

$$\text{Problem LSI: Minimize } \|y - Ax\| \text{ subject to } Gx \le h \tag{92}$$

One way to find the solution is to first recognize two important special cases. The Non-Negative Least Squares (NNLS) problem is a LS problem which requires all coefficients to be positive.

$$\text{Problem NNLS: Minimize } \|y - Ax\| \text{ subject to } x \ge 0 \tag{93}$$

The Least Distance Programming (LDP) problem is a minimization with respect to the origin.

$$\text{Problem LDP: Minimize } \|x\| \text{ subject to } Gx \le h \tag{94}$$

Any LSI problem can be changed to a LDP problem by the appropriate coordinate transformation. Similarly, a LDP problem can be solved with a NNLS algorithm. Thus, an algorithm solving the NNLS problem is the heart of the solution to the general LSI problem. The Kuhn-Tucker conditions are used in the NNLS algorithm to check whether a minimum is obtained.

Lawson and Hanson (1974) present a NNLS algorithm consisting of two loops. In the outer loop, the variables are brought into the problem, one at a time. The variable which would have the most positive coefficient is chosen. If the other coefficients remain positive, the loop repeats. This continues until all variables are included. If, at some point, one of the coefficients becomes negative, the inner loop starts. The inner loop adjusts the step direction to keep the coefficients nonnegative. Each time the inner loop is called, one of the coefficients is driven to zero. Thus a finite number of iterations of the inner loop are required. Further, the outer loop strictly decreases the residual norm at each iteration. Since there are a finite number of variable combinations which can be considered by the outer loop, the NNLS algorithm must converge.

A LDP problem is solved by forming a NNLS problem out of the constraints. The constraints are placed into an A matrix and y vector by

$$A = \begin{bmatrix} G' \\ \overline{h'} \end{bmatrix}$$
$$y = \begin{bmatrix} 0 \\ \overline{1} \end{bmatrix} \tag{95}$$

where in this case A is a $(n+1) \times r$ matrix and y is a $(n+1)$-vector. Then, solve a NNLS problem which minimizes $\| y - Ax \|$ subject to $x \geq 0$. Each column of A defines a boundary line within the feasible region. Figure 19 (Lawson and Hanson, 1974) illustrates a three constraint LDP problem. The solution point is the minimum euclidean distance from the origin that lies within the feasible region.
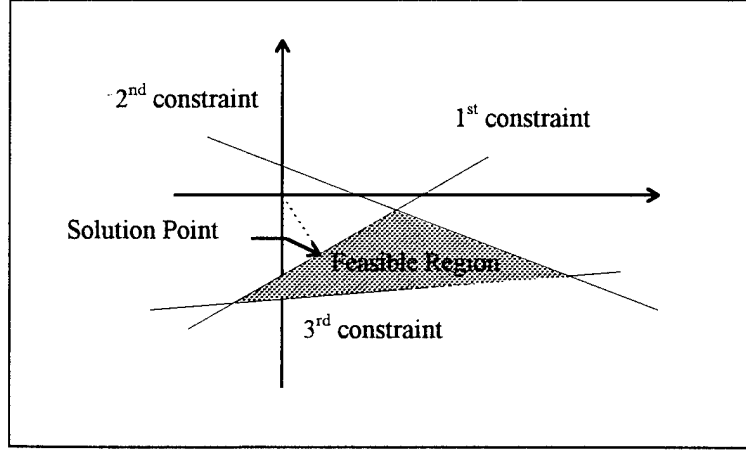
Figure 19: Least Distance Programming (LDP) Problem Illustration

Finally, a LSI problem can be converted to a LDP problem by a coordinate transformation. Use an orthogonal decomposition on the original LSI A matrix.

$$A = USV' = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1' \\ V_2' \end{bmatrix} \tag{96}$$

where A is an $m \times n$ matrix with rank $(A) = k$. Then, U is $m \times m$ orthogonal, S is $k \times k$ nonsingular, and V is $n \times n$ orthogonal. If a singular value decomposition is used, S is a $k \times k$ diagonal matrix containing the singular values of A. The partitions of U and V are consistent with the size of S, e.g., V_1' is $k \times n$.

The original LSI problem

$$\text{Problem LSI: Minimize } \| y - Ax \| \text{ subject to } Gx \leq h \tag{97}$$

is converted into the following LDP problem (Lawson and Hanson, 1974),

$$\text{Problem LDP: Minimize } \| z \| \text{ subject to } GV_1 S^{-1} z \leq h - GV_1 S^{-1} U_1' y \tag{98}$$

The original coordinates can be retrieved by

$$x = V_1 S^{-1} \left(z + U_1' y \right).$$

(99)

If one has inequality and equality constraints, the equality constraints are eliminated using the methods discussed earlier. Use an orthogonal change of variables to decompose the constraints, e.g. (79), to transform the system matrices as follows

$$\begin{bmatrix} C \\ A \\ G \end{bmatrix} V = \begin{bmatrix} C_1 & 0 \\ A_1 & A_2 \\ G_1 & G_2 \end{bmatrix}.$$

(100)

Now, the p-vector w_1 is the solution to

$$C_1 w_1 = d$$
$$w_1 = C_1^{-1} d$$

(101)

and the $(n\text{-}p)$-vector w_2 solves the LSI problem

$$\text{PROBLEM LSI: Minimize } \left\| A_2 w_2 - \left(y - A_1 w_1 \right) \right\|$$
$$\text{subject to } G_2 w_2 \leq h - G_1 w_1$$

(102)

Finally, transform back to the original variables using

$$x = Vw = \begin{bmatrix} V_1 & \vdots & V_2 \end{bmatrix} \begin{bmatrix} w_1 \\ -- \\ w_2 \end{bmatrix}.$$

(103)

Figure 20 summarizes the solution sequence for any LS problem. The fully constrained LS problem is solved via the following steps, a) decompose the constraints and reduce the dimensionality of the problem with (100) and (101), b) solve the remaining LSI problem (102) by forming a LDP problem (98), c) solve the LDP problem by changing it into a NNLS problem (95), d) finally, transform the solution back to the original variables with (103). While the solution description is cumbersome, the algorithm is readily coded as subroutines on a computer.

59

```
                    ┌─────────────────────────┐
                    │          LSG            │
                    │  Least Squares General  │
                    └─────────────────────────┘
                 ┌───────────┴───────────┐
    ┌────────────────────────┐   ┌──────────────────────────┐
    │          LSE           │   │           LSI            │
    │  Least Squares Equality│   │  Least Squares Inequality│
    └────────────────────────┘   └──────────────────────────┘
                                            │
                              ┌──────────────────────────────┐
                              │             LDP              │
                              │ Least Distance Programming    │
                              └──────────────────────────────┘
                                            │
                              ┌──────────────────────────────┐
                              │            NNLS              │
                              │ Non-Negative Least Squares    │
                              └──────────────────────────────┘
```
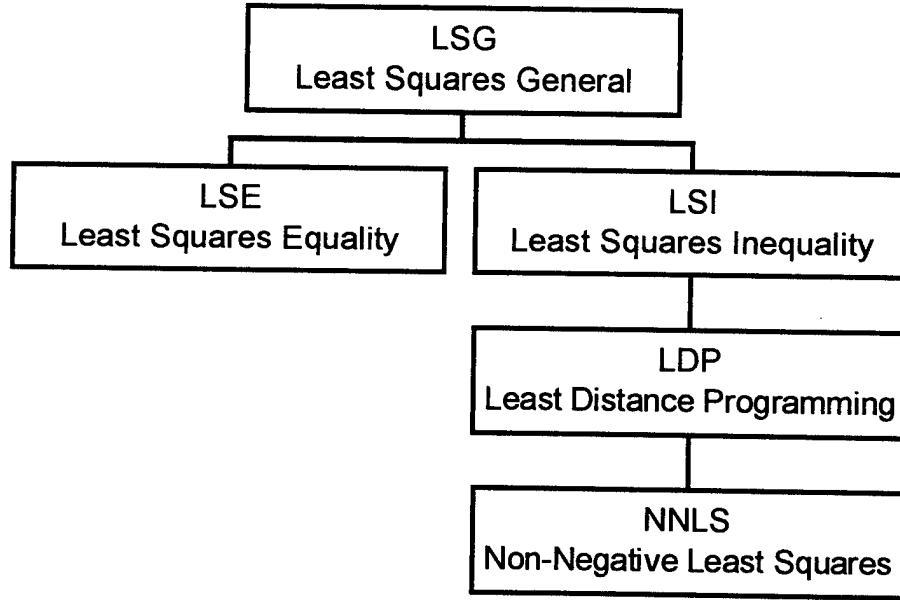
Figure 20: Solving the General Least Squares Problem

Earlier implementations used an alternate approach to solving the inequality problems (Gross and Schott, 1996a, 1996b). The gradient projection was used because of its more intuitive nature (Appendix B). The orthogonal decomposition coupled with the NNLS algorithm was eventually chosen over the gradient projection because of its better numerical stability.

Now that the mathematical foundation has been established, the remainder of this section describes how these tools are used to solve the unmixing and sharpening LS problems.

4.2 Unmixing: Over-Determined Least Squares

The spectral unmixing model from the previous chapter

$$LRXS_i = \sum_{e=1}^{n} R_{i,e} f_e + \varepsilon_i \tag{104}$$

is in the same form as the LS predictive model

$$y = \xi_0 + \sum_{i=1}^{n} A_i \xi_i + \varepsilon . \tag{105}$$

The constant coefficient, ξ_0, could be used to model shade or topography effects. However, since ξ_0 should be small, better results are obtained by forcing it to be zero. If a shade term is needed, it can be added as an endmember in the library. Equating the variables, this model is solved as an over-determined least squares problem,

$$\min\|y - \hat{y}\| = \min\|y - Ax\| \tag{106}$$

where $y = LRXS$ is an m-vector of digital counts from the spectral sensor, $A = R$ is an m x n matrix consisting of library reflectance values (e.g., endmembers), and $x = f$ is an n-vector of endmember fractions. If there are three endmembers and m spectral bands, the equation looks like

$$y = Ax$$
$$\begin{bmatrix} LRXS_1 \\ LRXS_2 \\ \vdots \\ LRXS_m \end{bmatrix} = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ \vdots & \vdots & \vdots \\ R_{m,1} & R_{m,2} & R_{m,3} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{107}$$

Low resolution equality constraints would be written in matrix form as

$$d = Cx$$
$$[1] = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{108}$$

The unconstrained solution is

$$x = (A'A)^{-1} A' y. \tag{109}$$

The constrained fractions are found via the Lagrange multiplier augmented system (77), or the decomposition technique. The difficulty arises in determining which n endmembers from the library to include in the A matrix.

A stepwise regression algorithm using a sequential F-test is used to add and remove candidate terms (endmembers) from the model. First, all possible one endmember models are formed. The model which best explains the measured digital counts (in reflectance units) will have the largest

MS(regression), the smallest MS(residual), the smallest squared error, and the largest F-to-enter ratio. If F-to-enter is greater than a threshold, the endmember is entered into the model.

Then, all the two endmember models which contain the entered material are examined. F-to-enter values are calculated for the (L - 1) candidate variables remaining in the library. If the largest F-to-enter is greater than the threshold, the appropriate endmember is added.

Once three variables have been entered into the model, it is possible that one may be removed. For example, although the first variable may have been highly correlated with the measurements, once the second and third variables have been entered, the first may become redundant. F-to-remove values are calculated for each of the entered endmembers as though each was the last to enter the regression. If the smallest F-to-remove is less than a threshold, that variable is removed from the model.

Although based upon an F-statistic, it is not necessary to use a value from an F-table. Since, the ratio used in the test is only approximately F distributed when the model is correct, a constant threshold value is used. The threshold for entering and removing variables are set equal to each other to prevent cycling. When a variable is neither entered nor removed, the algorithm terminates.

One question is how to incorporate the constraints into the stepwise regression algorithm. The literature is particularly silent on this point. Two options were considered. In the first, each time coefficients are calculated, they are subjected to the appropriate constraints. However, since the ANOVA analysis is intended for unconstrained regression, this likely violates some of the underlying assumptions. The second option is to use an unconstrained stepwise algorithm to determine the endmembers to be included in the model for each pixel. Then, the constraints are applied to those endmembers, and the coefficients recalculated (Dougherty, 1996).

An alternative approach calculates all possible subsets for a fixed number of materials (for example, five endmembers per pixel). An ad hoc complexity penalty is added to prevent overfitting extra materials when fewer would be appropriate. (Recall adding terms will always reduce the error.) The stepwise algorithm is preferred over the ad hoc subset method because it runs much quicker. One danger with the automated stepwise algorithm is that it may generate unrealistic models. A sanity check is performed on the per pixel fractions to flag inappropriate results. For example, large positive or negative

fractions indicate a model which could not possibly be correct. Once bad results are flagged, an alternate model can be calculated.

The alternate model is selected by temporarily removing from the library the most abundant material from the poor model, and rerunning the stepwise selection procedure. This forces the solution into a markedly different set of materials. Since the first result was so poor, a significantly different solution is desired. The combination of a stepwise regression, checking the fractions, and repeating the regression if necessary yields excellent results in a reasonable run time.

4.3 Sharpening: Under-Determined Least Squares

The high resolution model from the previous chapter

$$HRP_j = \sum_{e=1}^{n} R'_{pan,e} f_e^j + \varepsilon_j; \quad j = 1...s \tag{110}$$

is also cast in the form of a mixture problem. Recall there were n endmembers found by unmixing the corresponding low resolution superpixel, and we seek the unknown fractions in the s high resolution subpixels. For convenience, we can define x as an $n*s$-vector which contains the high resolution fractions,

$$x = \begin{bmatrix} f^1 \\ \overline{f^2} \\ \vdots \\ \overline{f^j} \end{bmatrix} \tag{111}$$

where the f^j are n-dimensional fraction vectors in the corresponding subpixels.

Just as in unmixing, the sharpening problem is in the same form as the LS predictive model

$$y = \xi_0 + \sum_{i=1}^{n} A_i \xi_i + \varepsilon. \tag{112}$$

Equating the variables, this model is solved as an *under*-determined least squares problem,

$$\min \|y - \hat{y}\| = \min \|y - Ax\| \tag{113}$$

63

where $y = HRP$ is an s-vector of digital counts from the spatial sensor, $A = R$ is a s x $(n*s)$ matrix consisting of library reflectance values in the sharpening band(s), and x is an $(n*s)$-vector of endmember fractions defined in (111). As an example, for $s = 4$ subpixels in a superpixel and $n = 3$ endmembers found from low resolution unmixing, the high resolution equation is written as

$$y = Ax$$

$$\begin{bmatrix} HRP_1 \\ HRP_2 \\ HRP_3 \\ HRP_4 \end{bmatrix} = \begin{bmatrix} R_{p,1} & R_{p,2} & R_{p,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & R_{p,1} & R_{p,2} & R_{p,3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & R_{p,1} & R_{p,2} & R_{p,3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_{p,1} & R_{p,2} & R_{p,3} \end{bmatrix} f' \quad (114)$$

where R_p is used instead of R_{pan}, and f' is not expanded so that the equation fits on the page.

Since there are more unknowns than equations, rank $(A) \leq s$, and $A'A$ is *not* invertible. There are an infinite number of least squares solutions. The pseudoinverse gives a solution which is best in the least squares sense, as well as providing a solution vector of minimum length (Gelb, 1974). The solution is

$$\begin{aligned} x &= A^{\#}y \\ &= A'(AA')^{-1}y \end{aligned} \quad (115)$$

Each of the three constraint conditions include consistency constraints equating the average of the high resolution fractions to the low resolution unmixing solution. Therefore, even the "unconstrained" sharpening case must be solved as a LS problem with equality constraints. Since $A'A$ is not invertible, forming an augmented matrix of the first order necessary conditions will not suffice. The orthogonal decomposition technique will work, since it does not require inverting the matrix.

Continuing the example of $s = 4$ subpixels in a superpixel, and $n = 3$ materials found by unmixing, the consistency constraints would be written as

$$\begin{bmatrix} 4f_1 \\ 4f_2 \\ 4f_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_1^1 \\ f_2^1 \\ f_3^1 \\ f_1^2 \\ f_2^2 \\ f_3^2 \\ f_1^3 \\ f_2^3 \\ f_3^3 \\ f_1^4 \\ f_2^4 \\ f_3^4 \end{bmatrix} \qquad (116)$$

Recall the subscript on f indicates the material, while the superscript indicates the subpixel number. When f has no superscript, it indicates the low resolution fraction. The equality constraints are

$$d = Cx$$

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} f' \qquad (117)$$

The fourth equality constraint

$$[1] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} f' \qquad (118)$$

is not necessary. It can be written as a linear combination of (116) and (117). Solving the fully constrained sharpening problem is the most challenging. Since there are $n*s$ unknowns, there are $2*n*s$ inequality constraints to be considered. The inequality constraints are written so the fraction is less than or equal to a constant. Therefore, $f \geq 0$ must be written as $-f \leq 0$.

$$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \geq \left[\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 1 \\ \hline -1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & -1 \end{array} \right] f' \qquad (119)$$

The number of constraints in each partition of (119) is consistent with the dimension of f'.

If the high resolution measurements are equal, corresponding to no high resolution spatial information, the high resolution fractions should equal the low resolution fractions. Thus, sharpening an aggregate mixture should have no effect. If there is spatial information contained in the spatial sensor data, the sharpening algorithm adjusts the high resolution fractions to return the minimum length vector with the least square residual error. Material maps of areal mixtures can be improved by sharpening.

This fusion algorithm was coded and tested using Matlab Version 4.2c (The MathWorks, Inc., Natick, MA) on a Unix platform. This choice was one of convenience; any programming language could have been used. The next chapter quantifies the performance of the fusion algorithm.

5. Results

This section describes the quantitative results obtained from unmixing and sharpening a synthetic test image. First, the experimental design is described, including the description of the data sets and the error metric. Then, sharpening results are presented. To separate the sharpening step from unmixing, the synthetic truth data was used to generate "perfectly unmixed" material maps. Sharpening these perfect maps quantifies the effect of the sharpening stage independently of the unmixing step. The next group of results quantifies the performance of the stepwise regression unmixing algorithm. The proposed unmixing algorithm computes the appropriate endmembers to use for each individual pixel. It is compared to traditional unmixing, where the same endmembers are used throughout the image. Finally, image fusion results are shown which combine the unmixing and sharpening algorithms. The section ends with a qualitative demonstration of the unmixing algorithm on a real multispectral image.

5.1 Experimental Design

5.1.1 Synthetic Imagery Characteristics

The algorithm development work capitalizes on the benefits of synthetic image generation (SIG). SIG scenes can be built with varying degrees of complexity. The ability to incrementally increase realism gives feedback to algorithm designers. They may test their designs under increasingly difficult conditions, and modify the designs to incrementally improve robustness. Secondly, since SIG is entirely computer created from a defined data source, the underlying "truth" is known. Algorithms can be evaluated under various conditions, where their performance can be quantified and compared to alternate techniques. After the algorithm is shown to work under simulated conditions, is can be tested on real imagery. Preliminary results using a small image scene are described in Gross and Schott (1996b).

A synthetic test image was generated with the DIRSIG algorithm (Schott et al., 1992). The bands were selected to simulate the Environmental Research Institute of Michigan (ERIM) M-7 airborne line scanner (MUG, 1995). The M-7 sensor has fifteen bands in the visible and infrared spectral regions. Figure 21 shows the band passes for the various spectral bands, while the numerical data are listed in

67

Appendix C. Data for three sharpening bands were also generated. To save processing a new image, representative M-7 bands at higher spatial resolution were used to sharpen. The visible sharpening band was a weighted average of M-7 bands 4 and 6. This represents a visible panchromatic band from 460 - 720 nm. Band 9 (760 - 1045 nm) was used as a near infrared (NIR) sharpening band, while band 13 (1400 - 1890 nm) served as a short wave infrared (SWIR) sharpening band. The nominal ground sample distance corresponding to a pixel in the base image was 1 meter.
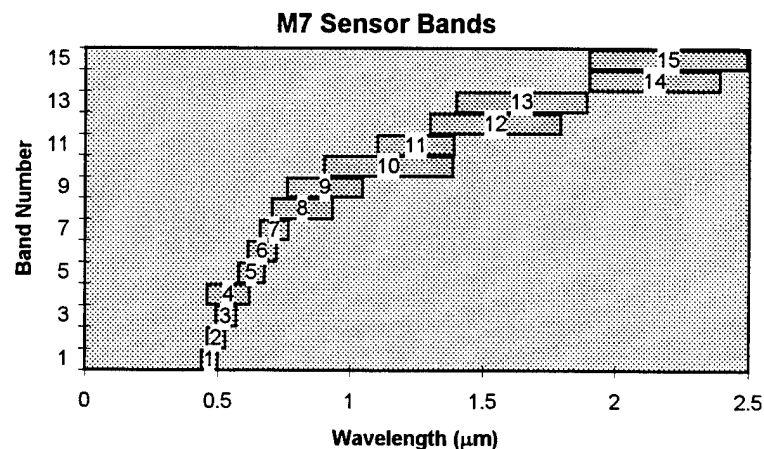
Figure 21: M-7 Sensor Band Passes

Band four (460 - 620 nm) of the test image is shown in Figure 22. The region contains deciduous forest, grass, a dirt road, a small lake, and several tanks and trucks. The vehicles consist of various materials, including camouflage paints, painted steel, canvas, and rubber. A spectral library containing ten materials was used for unmixing. Appendix C contains a detailed description of the spectral library used for the M-7 spectral and sharpening bands.
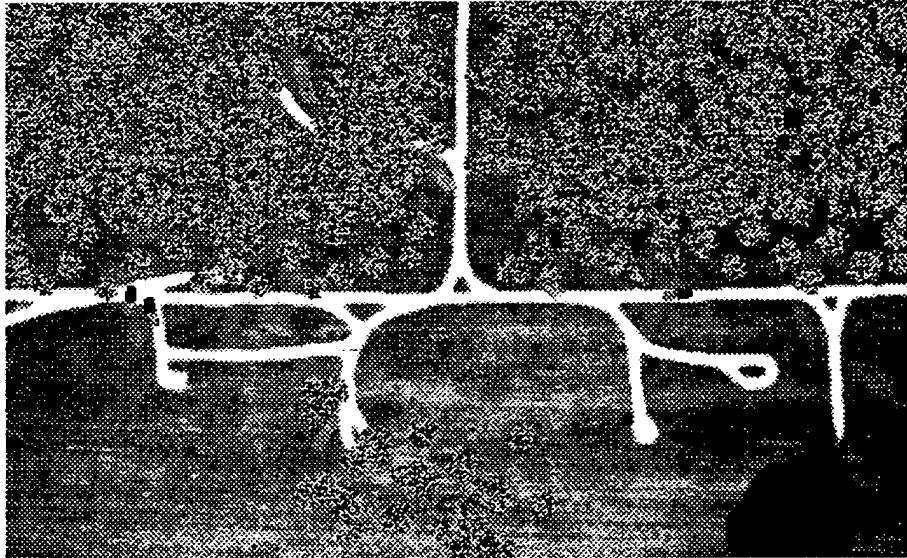
Figure 22: Band 4 (460 - 620 nm) of Synthetic Test Image

5.1.2 Data Sets

The DIRSIG algorithm uses a ray tracer. For each pixel in the image, a ray is cast to the appropriate point on the ground. By definition, each ray strikes a single material. Thus the original image contains only *pure* pixels. For this scene, the original scale was set at 1 meter per pixel (m/p). Aggregating (averaging) to a larger scale results in an image containing a smaller number of pixels for the same ground area. However, many of these averaged pixels are *mixtures* of several materials.

It is necessary to create fraction maps at a consistent scale in order to compare the results. Using averaging, images of decreased spatial resolution in the appropriate spectral bands were created at 2, 4, 8, 16, and 32 m/p. For example, the 4 m/p aggregation is a convenient high resolution scale. At this scale, the truth data contain fractions of materials in multiples of one sixteenth. Starting with the 1 m/p image cube and a 1 m/p material map, test image cubes, sharpening bands, and truth maps can be created at any lower resolution. Figure 23 is a block diagram showing how the synthetic image set is created. For clarity, only a few of the scales and data sets are shown. The material truth maps at any resolution are used to calculate "perfectly unmixed" material maps. These are used as inputs to the sharpening algorithm, allowing its performance to be measured independently of unmixing.
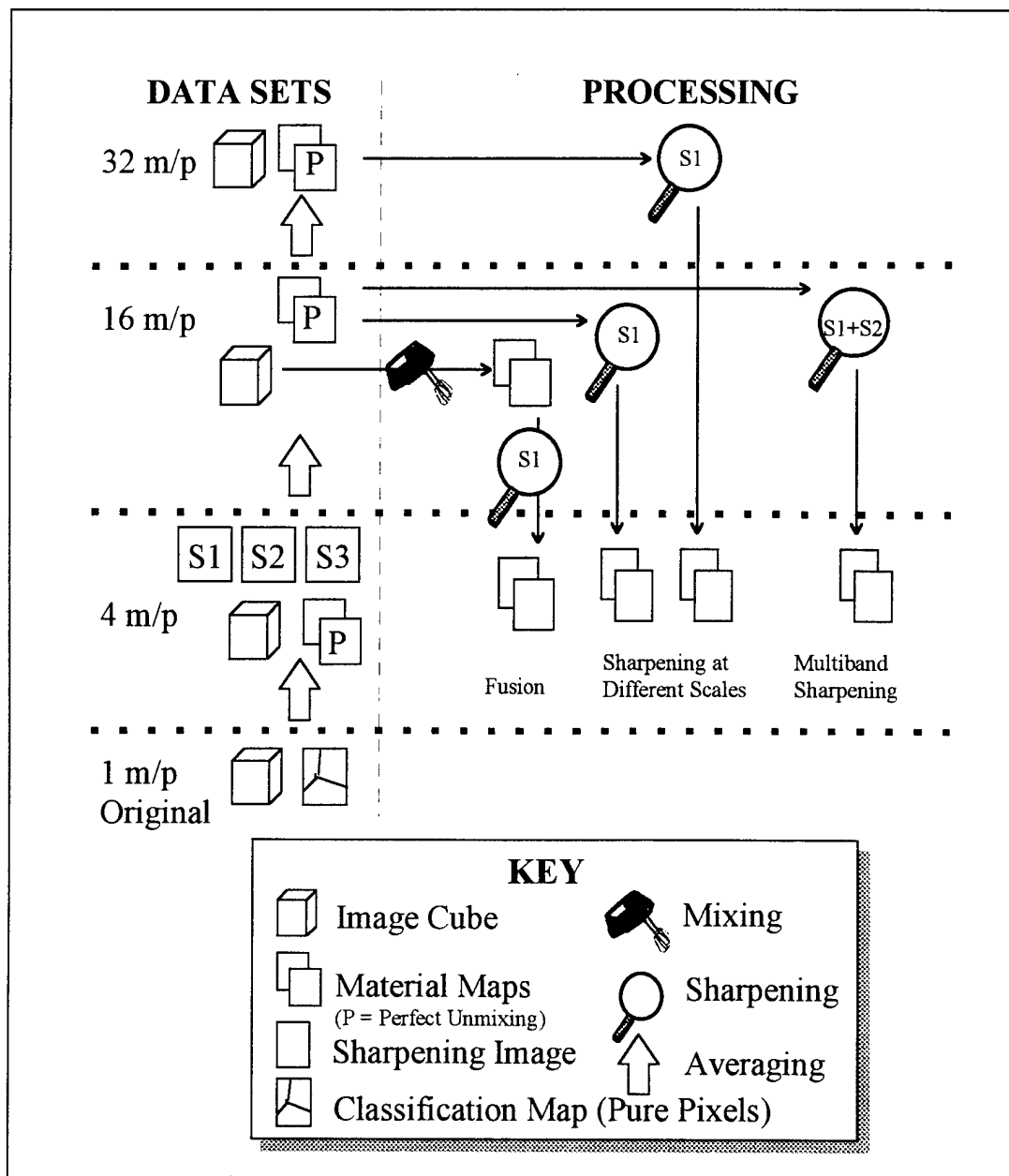
Figure 23: Creating SIG Data Sets

In Figure 23, the vertical axis indicates resolution. The unmixing operation (represented by the mixer icon), transforms image cubes into material maps in a horizontal (same scale) direction.

Sharpening, represented by a magnifying glass icon, operates down, increasing the resolution of the material maps. Different combinations of data, operations, and reference fractions (perfectly unmixed maps) are used to analyze the algorithms. Only a few examples are depicted in Figure 23. Fusion is shown as unmixing and then sharpening. Sharpening performance at different scales is represented by taking perfectly unmixed maps through the sharpening algorithm. Finally, multiband sharpening is indicated by the labels inside the magnifying glass. A complete list of the data sets generated is included as Appendix D.

Creating the reference data sets illustrated in Figure 23 without SIG would require extensive ground surveys and perfect registration of image data with ground reflectance samples to ensure pure pixels.

Some of the perfectly unmixed material maps at the 4 m/p scale are shown in Figure 24. Recall the intensity (gray level) in a material map is proportional to the fraction of that material at each pixel location. At this scale, there are 17 gray levels possible, corresponding to fractions between zero and one.

Deciduous Trees   Grass

Dirt   Lake Water
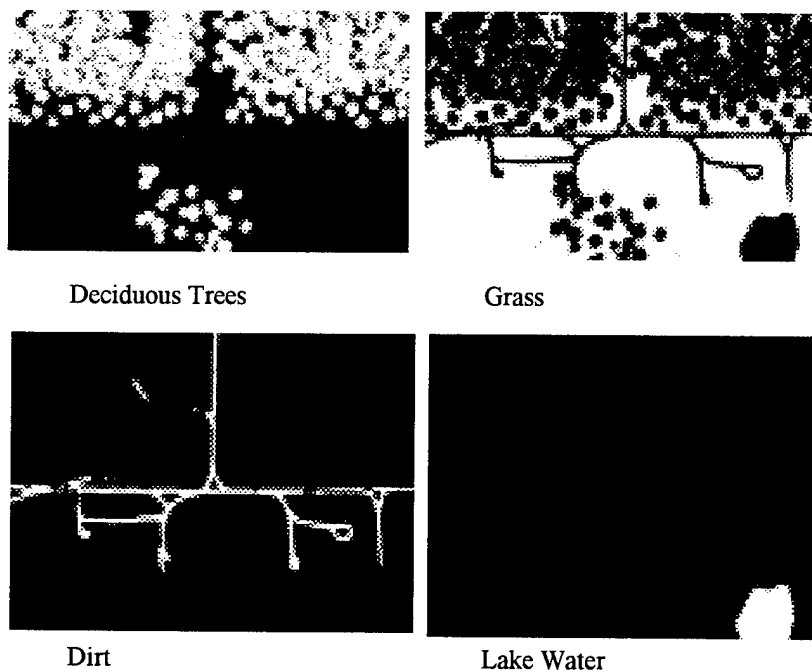
Figure 24: Perfectly Unmixed Material Maps (4 m/p)

5.1.3 Error Metric

To capitalize on the truth data available from the SIG imagery, a single error metric is used to compare the results. A squared error (SE) per pixel was calculated for each set of fraction maps,

$$SE = \frac{1}{N} \sum_{pixels} \sum_{materials} \left(f_{truth} - f_{test}\right)^2 \qquad (120)$$

The summation over the pixels includes the entire region of interest (N pixels). The material summation is over the entire library of materials. The relative magnitude of this error metric gives a convenient measure of the match between the test fractions and the truth fractions. All errors of commission and omission are penalized by this metric.

5.1.4 Comparing Maps at Different Scales

By definition, the perfectly unmixed data sets have no error. However, there is an opportunity cost due to not sharpening. A method is required to quantify the effect of leaving the material maps at low resolution. Pixel *replication* was used to compare maps of different scales.

In pixel replication, the fractions of each material are repeated for each subpixel. For example, to replicate from a 16 m/p scale to a 8 m/p scale, each 16 meter pixel fraction is replicated into a 2 x 2 array of 8 meter pixels. Each of these four values can then be compared to the reference values at 8 m/p. If the pixel is pure at the 16 meter scale, replication has no effect. However, if the material concentrations are spatially separable at 8 m/p (an areal as opposed to an aggregate mixture), replicating results in error. This is the error due to *not* sharpening.

Figure 25 shows the error that arises from replicating the perfect material maps. No algorithms (unmixing or sharpening) are applied to this data. The graph shows that the amount of replication error depends upon the ratio of the scales, not the initial or final scale.
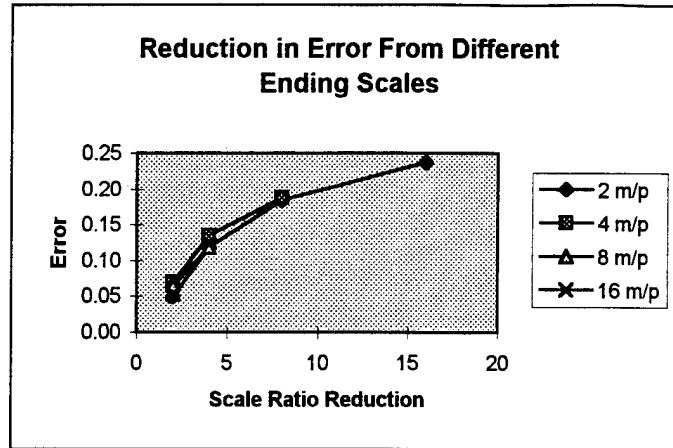
Figure 25: Comparing Maps at Different Scales

5.2 Sharpening Results

To analyze sharpening independently of unmixing, the sharpening algorithm was applied to the perfectly unmixed data sets. Perfectly unmixed material maps are derived directly from the base image classification map. The fractions in the perfect maps are multiples of the aggregation scale used to form the maps.

In the following subsections, different aspects of sharpening performance are highlighted. The data were generated for each of the various constraint conditions. The legends on the charts indicate which constraint condition was applied. "None" corresponds to the unconstrained algorithm, "partial" means the fractions were required to sum to unity, and "full" signifies the additional prohibition of negative fractions or fractions greater than one.

5.2.1 Band Selection

Figure 26 compares the individual use of three sharpening bands. The reference data are the perfect material maps at the 4 m/p scale. The initial scale for these data was 16 m/p, so the superpixels contain arrays of 4 x 4 subpixels. The line on the graph represents the replication error in going from the 16 m/p to the 4 m/p scale.

73
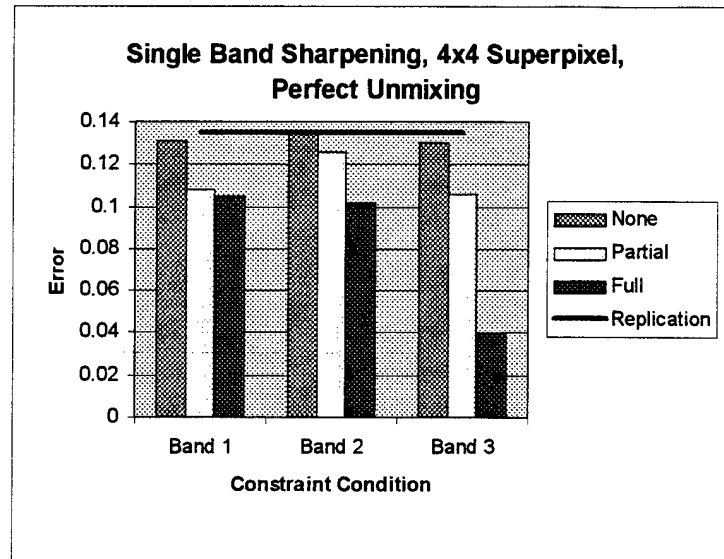
Figure 26: Single Band Sharpening

The figure shows that in all cases sharpening results in less error than replicating (i.e., not sharpening). In other words, the high resolution maps resulting from sharpening perfectly unmixed material maps are better than the perfect low resolution maps.

Figure 26 also shows that, for perfect unmixing, partially and fully constrained sharpening are better than unconstrained. Constraining the fractions reduces the error. Using band 3 in the fully constrained algorithm was able to explain a great deal of the error. The SWIR sharpening band variance is much greater than the visible band variance. Thus, there is more "information" in band 3.

Error sources in these synthetic images include texture variation in the material realizations, orientation differences of the individual objects in the scene, and differences between the image and library reflectance values. No sensor or atmospheric effects are included in the image, so calibration and atmospheric correction errors are absent.

Figure 27 shows results for multiple sharpening bands. Multiple sharpening bands are easily incorporated into the algorithm. They serve as additional measurements, and increase the reliability of the result. In addition, the bands measure different spectral regions. Choosing different spectral bands increases the likelihood of distinguishing between similar materials, thus resulting in more accurate fractions.
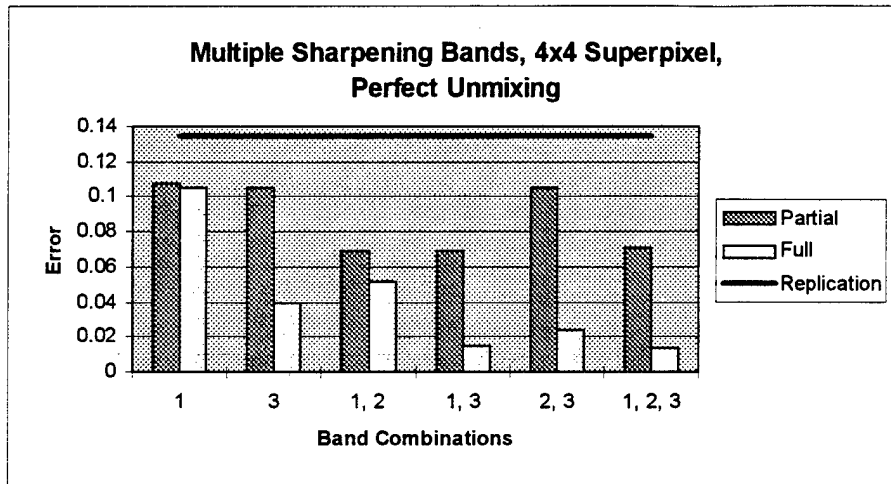
Figure 27: Multiple Band Sharpening

For comparison, the replication and some single band results are duplicated from Figure 26. Since the unconstrained results were poor, Figure 27 only shows partially and fully constrained performance. For this image, the best results occur when including band 3 in the fully constrained algorithm. However, if band 3 is present, including band 2 adds little additional value.

Despite the promising results with band 3, the remaining single band sharpening examples will use band 1 as the sharpening band. This is because the most likely sharpening applications will be to combine multispectral data with a visible panchromatic band. However, as these results indicate, if one could select a sensor with which to do image fusion, a SWIR band may have great value. Of course, the selection of a sharpening band would be dependent upon the materials in the particular scene under examination.

The ability to sharpen with multiple bands is a powerful tool. One could select sharpening bands that are decorrelated with each other to increase the amount of new information provided by the additional band(s). These bands provide extra measurements which can be used to distinguish similar materials. If desired, one can select sharpening bands that are correlated with key material parameters. The system designer could *choose* the sharpening bands in order to increase the overall fusion accuracy or to highlight particular materials. As future sensor systems become available, the utility of selecting the most effective sharpening bands could have a large payoff.

5.2.2 Scale

Figure 28 shows single band sharpening at three different scales. Larger superpixels have more error. Recall, the number of unknowns in the sharpening algorithm is $n*s$, where n is the number of materials, and s is the number of subpixels in the superpixel. Doubling the sharpening scale increases s by a factor of four. Further, by increasing the size of the superpixel, more materials are likely to be present. Thus, n will be larger as well. The larger number of unknowns results in more opportunity for error as well as longer run time, especially in the (iterative) fully constrained algorithm.
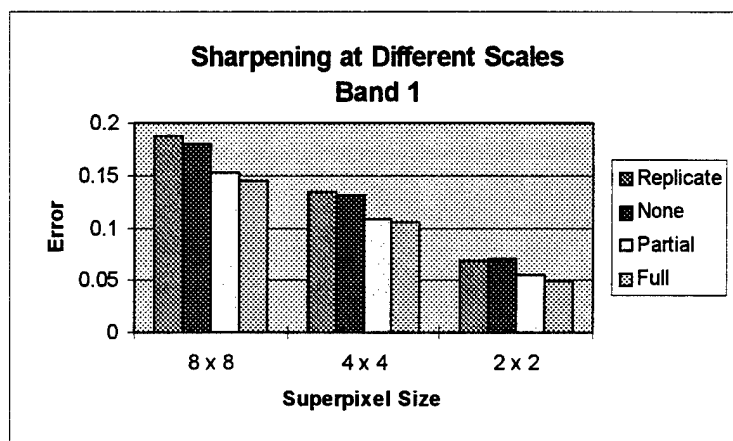
Figure 28: Sharpening at Different Scales

The bar charts in Figure 28 have similar shape. Unconstrained sharpening is not very different than simple replication. The partially and fully constrained algorithms are needed to improve the fractions. The poor performance of unconstrained sharpening can be explained by the large amounts of variation (e.g., texture) in the trees, grass, and dirt. Variation in the digital counts means a mismatch with the endmember library, and the algorithm drives the fractions away from the true values.

5.2.3 Variation

To examine the effects of variation, the SIG scene was regenerated without texture. In the DIRSIG algorithm, when a ray strikes a material, its reflective (emissive) properties are retrieved from a data base. For an image without texture, there is a single reflectance curve corresponding to each material. When texture is included, the algorithm selects from a large family of curves according to an

76

accompanying texture map. Hence, the digital counts for a particular realization of a material take on a random character, while maintaining spectral and spatial correlation with the rest of the image (Schott, et al., 1995). To create the spectral library, the reflectance for each material was calculated by averaging over the entire family of curves. The reflectance for each band was calculated by averaging this result over the bandpass.

Figure 29 presents three groups of plots. The first set, repeated from Figure 28, shows the error when texture variation is included in the image. The second set shows the result of using the original spectral library to sharpen an image with no texture. In the third set, the spectral library was recalculated to accurately reflect the characteristics of the materials without texture. The new library data is also included in Appendix C. The dirt in the original image was included as part of the grass texture. Therefore, without texture, there is no dirt in the scene. Thus, for the no texture and new library data, the perfect material maps were regenerated without the dirt material class.
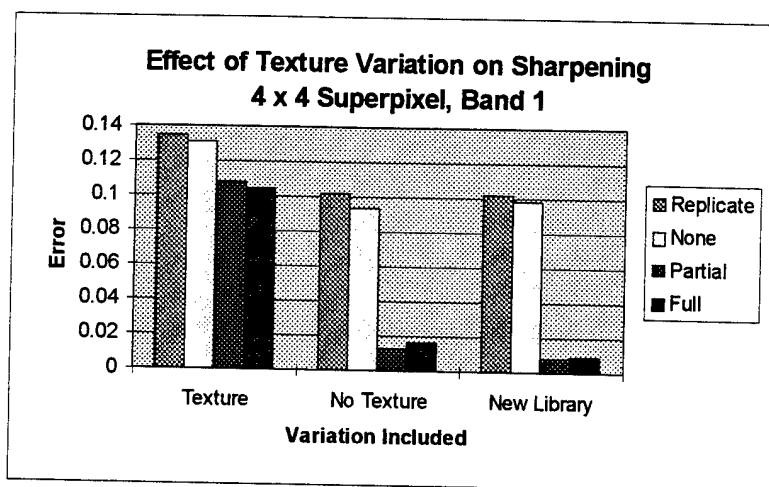
Figure 29: Effect of Texture Variation on Sharpening

From the figure one can conclude that sharpening is better than not sharpening (replication) in all cases. Partially and fully constrained results are much better than unconstrained. With little or no variation, sharpening can remove more than 90% of the error. Even with texture, sharpening gives a 20% improvement.

The poor performance of the unconstrained algorithm in the "no texture" and "new library" runs is harder to explain. Although texture was removed from the material realizations, the other variation

sources must still be significant. Even in the no-texture image, digital counts vary due to orientation effects. This would especially be evident in the large number of trees in the image. Trees consist of multiple "facets" at different orientations. In DIRSIG, reflectance can vary with orientation, so even a single tree curve results in many different digital count values. Unconstrained sharpening over-adjusts the fractions while it tries to match the "noise" due to variation.

5.3 Unmixing Results

The secondary objective of this research is to improve upon the spectral unmixing technique. A stepwise regression approach is used to select the endmembers on a pixel by pixel basis (Gross and Schott, 1996c). Traditional unmixing uses the same set of endmembers throughout the entire scene. By including excess terms in the model, the traditional approach can induce errors in the fractions. Unmixing with the wrong number of endmembers increases the error. Since the number of materials in each pixel varies across the scene, the stepwise regression algorithm offers an improvement.

Measuring the performance of an unmixing algorithm requires detailed knowledge of the materials in the scene. Synthetic imagery presents an opportunity to quantify unmixing performance. Unmixing is done at the "low" spatial resolution (relative to the sharpened result). One can measure unmixing accuracy against perfectly unmixed maps at the low resolution. Alternately, one can measure the combined effects of unmixing and not sharpening by comparing the unmixing result to the perfect high resolution material maps. The first comparison is adequate for analyzing the unmixing algorithm. The second is required if one wants to relate unmixing and sharpening.

5.3.1 Scale

Figure 30 compares unmixing results to perfectly unmixed maps at the same scale. Five values are reported at each scale. Option 1 for partial and full constraints indicates the constraints were included at each stage of the stepwise regression. Since the Analysis of Variance technique applies to unconstrained least squares problems, including the constraints violates some fundamental assumptions. This is verified by the results; option 1 yields poor fractions.

78

Option 2 results incorporate the constraints *after* stepwise regression. In other words, the stepwise algorithm is run unconstrained to determine the endmembers that minimize the error. Then, the constrained coefficients are recalculated *for that set of endmembers.*
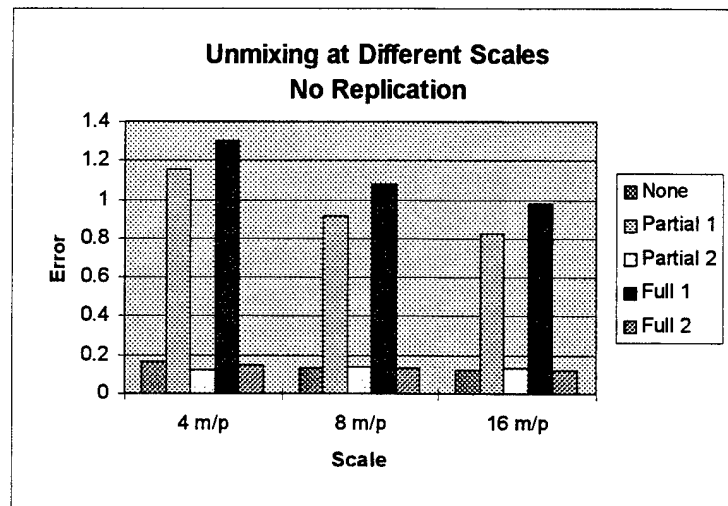
Figure 30: Unmixing at Different Scales

The results indicate unmixing error is unaffected by scale. This is not surprising since the unmixing model is an abundance-weighted average of the endmember spectra. The figure also implies the partially constrained solution may be just as good as the fully constrained algorithm. This would significantly aid implementation, since one could avoid checking inequality constraints. Recall a fully constrained algorithm must iterate to find the solution.

5.3.2 Benefit of Unmixing Each Pixel

Spectral unmixing results have been difficult to quantify due to a lack of ground truth. Because this test image is synthetically generated, a detailed material map is known. Thus it is possible quantify unmixing performance.

Traditional unmixing uses a fixed number of endmembers for the entire scene. Even if a certain material is only present in a small number of pixels, traditional unmixing calculates a coefficient (fraction) for that material at each pixel location. Including excess terms in the model induces errors in the remaining fractions.

Figure 31 compares unmixing results at the 8 meter scale for three different algorithms. In the first, traditional unconstrained unmixing is applied with the full 10 endmember library. In the second, four dominant endmembers are chosen to reduce the number of unnecessary coefficients. The third algorithm is the proposed unmixing scheme based upon stepwise regression. In the third algorithm, the number of endmembers is adjusted at each pixel. The stepwise algorithm selected the appropriate endmembers from a library of ten materials. Results from all three algorithms are compared to the perfect material maps at 8 m/p without replication. The algorithms were run unconstrained to be consistent with the unmixing implementation in the ENVI software package (Research Systems, Inc., 1995).
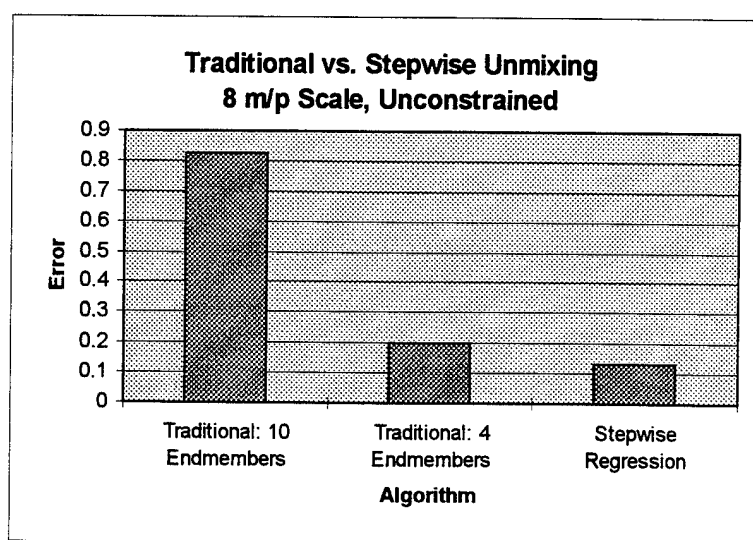
Figure 31: Traditional vs. Stepwise (Per Pixel) Unmixing

The results show the improvement of the proposed unmixing scheme over traditional methods. If one unmixes the entire scene with ten endmembers, the excess materials cause severe overfit in the fractions. This results in poor performance relative to ground truth. Note that 10 endmembers results in the *lowest* residual RMS error due to the model fitting the noise. RMS is not a good indicator of overall unmixing performance. (Spatial patterns in an RMS image, however, can indicate regions of unexplained materials.) Unmixing only the trees, grass, dirt, and water considerably reduces the error. However, by doing this, one is unable to map the vehicles. The stepwise algorithm reduces the overfit, constructs material maps for all materials in the image, and results in a 35% improvement in unmixing performance.

As a prelude to including sharpening, Figure 32 shows unmixing at 16 m/p and replication to 4 m/p. Note here the perfect unmixing error corresponds to replication error. Constrained unmixing was done according to option 2 discussed above. The endmembers are selected via unconstrained stepwise regression. Then the coefficients are recalculated with the appropriate constraints.
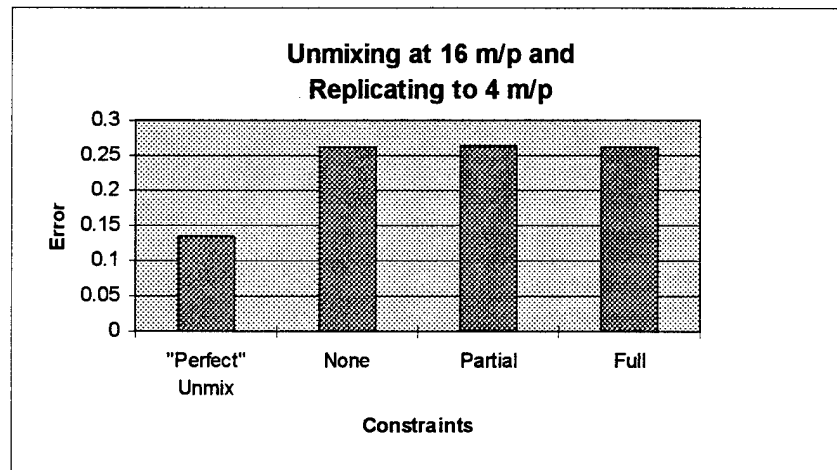
Figure 32: Unmixing and Replication to a Higher Spatial Resolution

The figure indicates unconstrained unmixing yields approximately the same error as constrained unmixing. It is unclear how this would extend to other images. Intuition suggests constraining the fractions would be appropriate. Perhaps the fraction checking incorporated into the stepwise regression algorithm was sufficient. In the figure, one also sees there is some error inherent in the unmixing algorithm due to mismatches between the image and the library. The additional error due to pixel replication is the potential improvement which can be obtained by sharpening. However, if unmixing selects the wrong endmembers, that error is unrecoverable as sharpening takes the unmixing solution as its starting point.

5.3.3 Variation

As with sharpening, variation due to texture makes unmixing more difficult. Figure 33 compares the error that remains after unmixing an image with texture to the error that occurs when the image has "no texture." As before, the new library data corresponds to a no texture image with an adjusted spectral library that matches the image data as closely as possible.

81

Effect of Texture Variation on Unmixing
No Replication, 16 m/p Scale
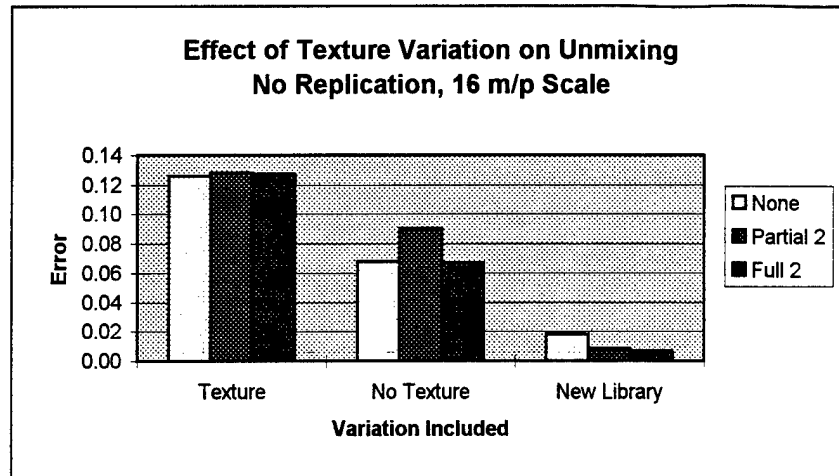
Figure 33: Effect of Texture Variation on Unmixing

Unmixing performance is better when the library reflectance matches the reflectances realized within the scene. The less variation in the image, the better the unmixing fit.

5.4 Fusion Results

In this section, unmixing and sharpening are combined in an image fusion algorithm. First, stepwise unmixing is applied to the low resolution image cube to make low spatial resolution material maps. Then the maps are sharpened with high resolution sharpening band(s) to give high resolution material maps. The previous section already showed stepwise unmixing is better than the traditional approach. The following results demonstrate that sharpening improves the material maps even further. Figure 34 shows representative material maps. The first column contains low resolution maps at a 16 m/p scale. The second column depicts the sharpened maps at 4 m/p.
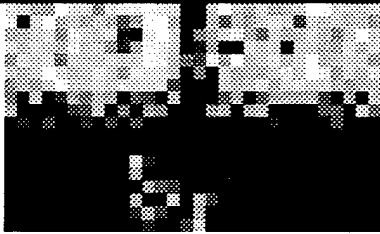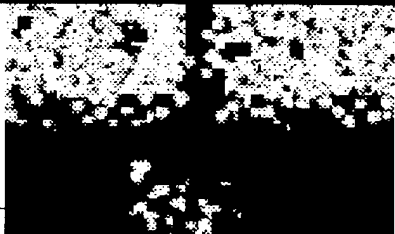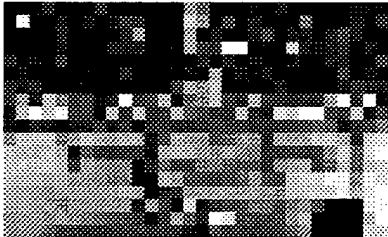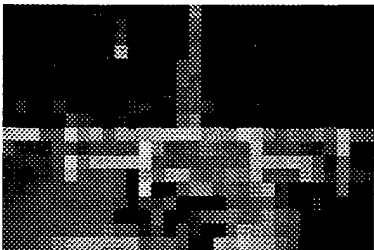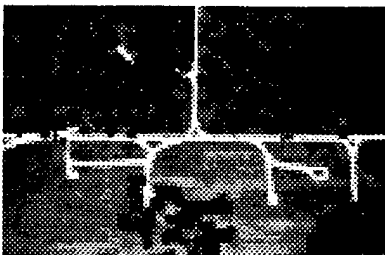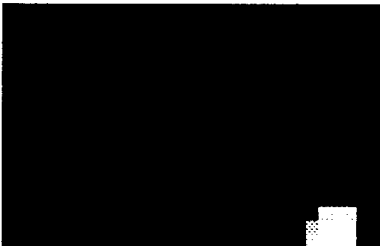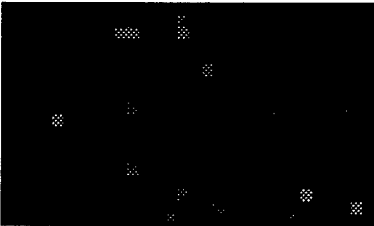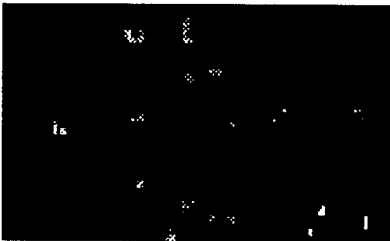
MATERIAL MAPS	UNMIXED (16 m)	SHARPENED (4m)

Trees

Grass

Dirt

Water

Camouflage Paint

Figure 34: Unmixed (16m) vs. Sharpened (4m) Material Maps

5.4.1 Single Band

In Figure 35, single band image fusion is compared to unmixing without sharpening. The best results are in the fully constrained case, although partially constrained fusion performance is similar. Sharpening gives approximately a 10% improvement in the error. Unmixing and sharpening is better than unmixing without sharpening.
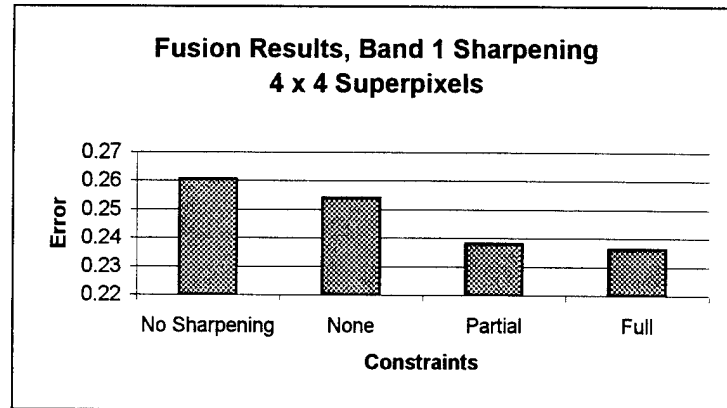
Figure 35: Fusion with a Single Sharpening Band

The improvement obtained using constrained fusion can be considered statistically significant with more than 99.9% confidence. The 3% improvement using unconstrained fusion is not statistically significant unless one makes unrealistic assumptions about the error distribution. The difference between partially and fully constrained fusion is also not statistically significant (Appendix E).

5.4.2 Multiple Band

Figure 36 shows the fully constrained case when using multiple sharpening bands. By adding the SWIR information in band 3, the error is reduced by 30%.

**Fusion with Multiple Sharpening Bands,
Fully Constrained**

Error vs *Sharpening Bands*: No Sharpening, 1, 1, 3, 1, 2, 3 (y-axis 0.00 to 0.30)
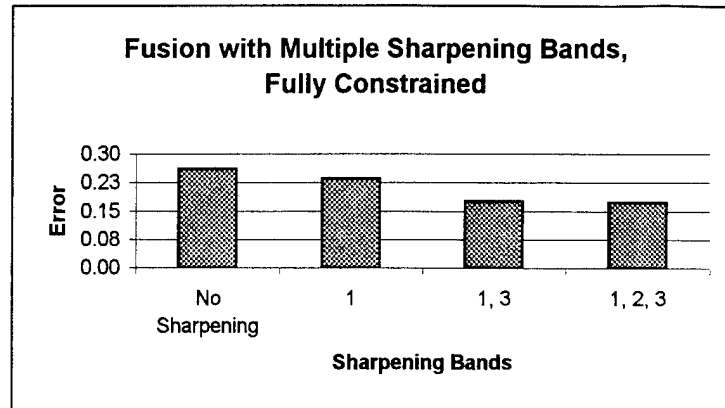
Figure 36: Fusion with Multiple Sharpening Bands

5.4.3 Effect of Texture Variation

Finally, the image fusion algorithm was applied to the no-texture image. The corrected new library was used to illustrate near ideal conditions. Figure 37 repeats the texture results for comparison. Obviously, the results are better without texture. However, the algorithm is still useful when applied to realistic imagery.

Effect of Texture Variation on Fusion

Error vs *Texture Condition*: Texture, No Texture (y-axis 0.00 to 0.30). Legend: No Sharpening, Band 1, Bands 1, 2, 3
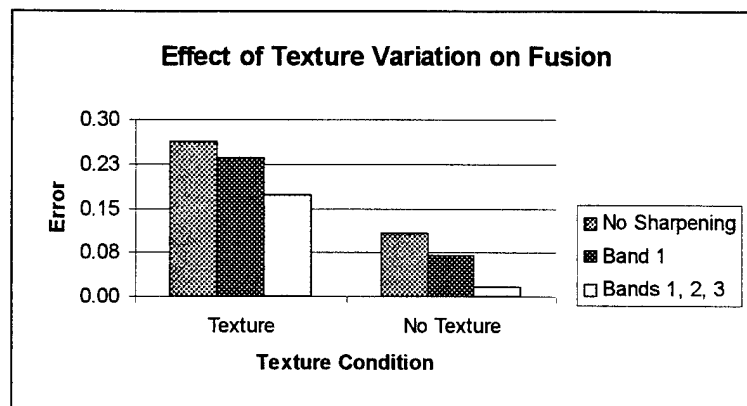
Figure 37: Effect of Texture on Image Fusion Algorithm

Table 4 summarizes sharpening performance. It shows the percent improvement relative to the stepwise unmixing algorithm. The percentages are calculated as the difference between the sharpened result and the replicated (no sharpening) result, divided by the no sharpening error. The results relative to perfect unmixing are repeated for completeness.

Sharpening with	Stepwise Unmixing		Perfect Unmixing	
	Texture	No Texture	Texture	No Texture
Single Band	10%	34%	22%	91%
Multiple Bands	34%	84%	90%	97%

Table 4: Percent Improvement of Sharpening Over Replication

It is not surprising that sharpening works better on perfectly unmixed maps. Since sharpening uses the endmembers found via unmixing, if unmixing selects the incorrect materials, sharpening will fail. Also, if the low resolution fractions are wrong, and the pixel contains an aggregate mixture relative to the sharpening bands, sharpening cannot improve the result. However, perfect unmixing is unattainable. These results show sharpening improves realistic maps created via stepwise spectral unmixing. Using more sharpening bands improves the fractions, while texture variation causes errors.

5.4.4 Spatial Error Distribution

The above results were derived using an error metric which collapses the performance of each algorithm into a single value. This masks any spatial variation in the error. Figure 38 plots the squared error in the material fractions at each pixel location for a well performing fusion algorithm. In this figure, bright pixels indicate large error, while dark pixels contain little or no error.

Figure 38: Spatial Error Distribution: High Resolution Maps vs. Truth

The brightest pixels in the figure correspond to the boundaries between the materials. Several factors influence the fractions at the boundaries. If the sharpening bands have low spectral contrast between the materials (e.g., an aggregate mixture), sharpening will be unable to properly place the high resolution fractions. If the unmixing algorithm selects a wrong endmember, or the algorithm halts before adding a required material, this induces unrecoverable errors in the sharpening stage. Recall that the sharpening constraints require consistency with the low resolution unmixing result. Finally, the trees contain increased texture and angular effects at the edges. All these factors combine to make it more difficult to sharpen mixed pixels.

A second source of error is the large grassy area at the bottom center of the image. In this area, the error arises from the truth maps, not the algorithm. In this DIRSIG test scene, the dirt road is implemented by manipulating the grass texture map. A material called grassy-dirt is used to create the original scene. For unmixing, separate grass and dirt materials are used, and fractions developed for each material. However, in this region of the image, the synthetic grassy-dirt actually has a mixed spectral nature. Recall the truth maps are generated by averaging the appropriate *pure* materials into perfect fraction maps. Since the underlying materials are not pure, the truth maps incorrectly state the grass and dirt fractions. In this region, the unmixing algorithm works better than the method for generating the truth data.

5.5 Summary

Although quantitative data is presented for only a single synthetic scene, some tentative conclusions can be drawn. Sharpening reduces error relative to unmixing alone. Since the stepwise unmixing algorithm was shown to be an improvement over the traditional method, this combination of unmixing and sharpening provides an significant advance in the creation of high resolution material maps.

Determining the best constraint condition to implement is not obvious. Unmixing results show similar performance under all three conditions. This may be because the stepwise regression algorithm

includes significant model checking. Constrained unmixing is best performed after the algorithm selects the endmembers.

Unconstrained sharpening, however, was significantly worse than constrained sharpening. The fully constrained algorithm was slightly better than partially constrained sharpening, however one suffers a large computational burden to get a small improvement in performance. The overhead associated with fully constrained sharpening arises because the algorithm must iterate through the inequality constraints. Algorithm run time is also strongly dependent upon the size of the superpixel. Sharpening a small image with 2x2 superpixels requires 2-3 minutes with the engineering code. Sharpening an 8x8 superpixel takes 7 minutes if unconstrained, 9 minutes when partially constrained, but over 3 hours when fully constraining the fractions. Although coding improvements could change the algorithm run times, fully constrained sharpening at large scales will be very computationally intensive.

Although the results are promising, some limitations are immediately evident. Since the amount of improvement was shown to be related to the amount of variation in the image, fusion results will be scene dependent. Clearly, the number of mixed pixels in the scene is related to the potential gain due to sharpening. Only pixels that are aggregate mixtures at low resolution, but areal mixtures at high resolution will be aided by sharpening. Finally, as the data with the new library highlighted, a priori knowledge of the spectral characteristics of the endmembers directly corresponds to better prediction of the fractions.

Using mean spectral vectors for the endmembers suggests slightly negative fractions and fractions slightly greater than one may be appropriate. This would indicate partial constraints may be more appropriate than full constraints. In this regard, this analysis was inconclusive. For the amount of variation in the test image, the fully constrained algorithm was the best. However, one clear result is that an unconstrained algorithm could be improved upon by adding constraints. Most of the value is gained by adding partial constraints. The fully constrained algorithm showed only a slight additional improvement. In the case where significant variation is present, the best solution may be a partially constrained result which allows slightly negative fractions.

88

5.6 Application to a "Real" Image

A real M-7 1 meter resolution image (Figure 39) was aggregated by a factor of ten in each direction to create an image cube at 10 meter resolution. To eliminate the need for calibration to reflectance units, this cube was unmixed with eight in-scene derived endmembers. To approximate ground truth, the original image was also aggregated by a factor of two in each direction to create a 2m cube. A few of the material maps are shown in the next two figures. Figure 40 contains 10m material maps. Figure 41 contains material maps unmixed at 2m resolution. Sharpening was not accomplished because of low confidence in the registration procedure.

Figure 39: Real M-7 Image, 1m Resolution

Asphalt

Dirt

Grass

Roof
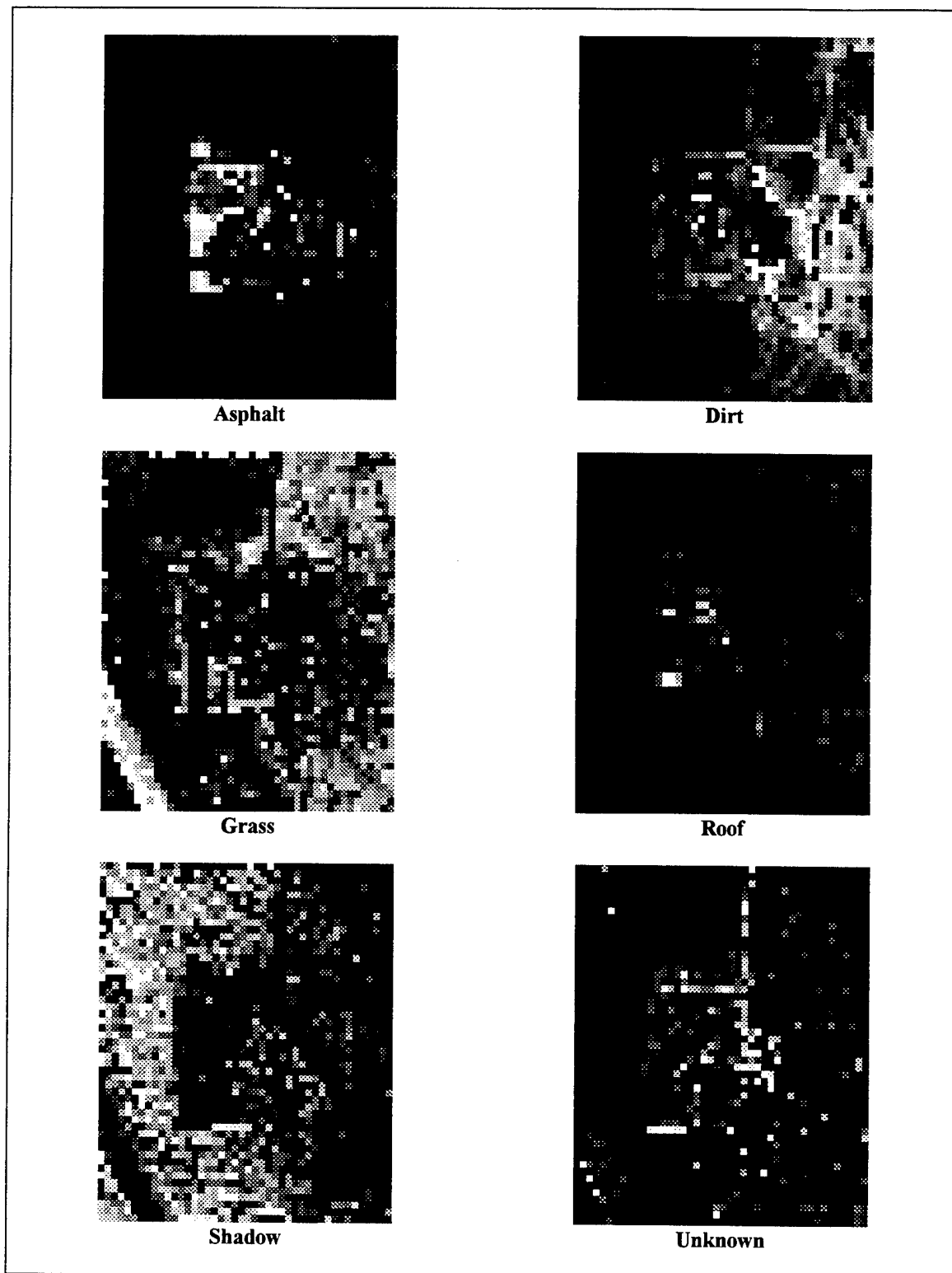
Shadow

Unknown

Figure 40: 10m Material Maps Using Stepwise Unmixing

90
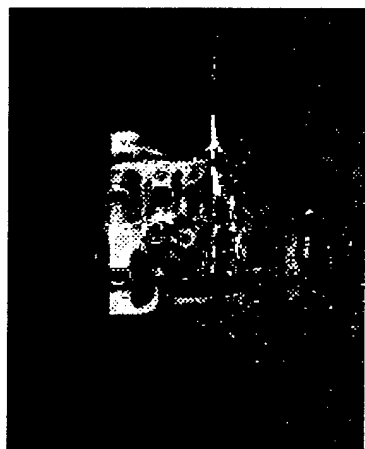
Asphalt

Dirt

Grass

Roof
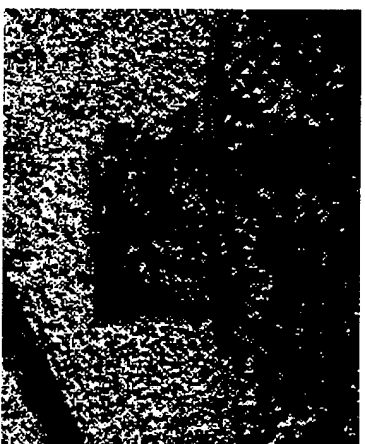
Shadow

Unknown

Figure 41: 2m Material Maps Using Stepwise Unmixing

Since there is no ground truth, the true material fractions are unknown. It is possible to compare the fractions obtained via stepwise unmixing with those obtained in the traditional manner. Figure 42 shows unmixing all 8 endmembers in each pixel has almost twice the error as the stepwise approach.

Traditional vs. Stepwise Unmixing Real Image, Unconstrained
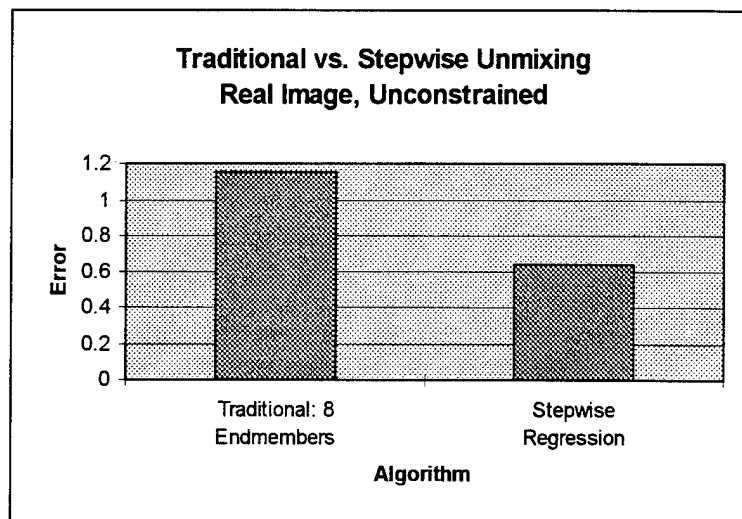
Figure 42: Unmixing a Real M-7 Image

Although Figure 42 attempts to quantify the error, one must recognize there is no ground truth data. The error values stated should be treated as relative errors. Note, in a typical unmixing implementation, one does not have a reference image. One should be aware of the dangers of using RMS fitting error as the metric. RMS error may or may not be related to the actual materials on the ground. Furthermore, RMS errors are always decreased by adding endmembers. RMS does not penalize overfit.

6. Conclusions

Both research objectives were satisfied. An image fusion algorithm based upon spectral mixture analysis was developed to spatially enhance material maps. The algorithm includes an improved spectral unmixing routine that uses stepwise regression to select the appropriate materials (endmembers) on a pixel by pixel basis. This is contrasted with traditional unmixing where a set of endmembers is used for the entire scene.

The second stage of the algorithm is called sharpening. The low resolution material maps produced in the unmixing step are sharpened with data from high spatial resolution sensors. The result is a set of high resolution material maps. A synthetic test image was used to quantify the performance of each part of the fusion algorithm. Using synthetic data provides complete knowledge of the ground truth, so that quantitative evaluation of material fractions may be made.

6.1 Contributions

The specific contributions of this research can be grouped into three areas. In a general sense, SIG data was essential in order to quantify algorithm performance. Spectral unmixing algorithms have been especially difficult to quantify because a detailed knowledge of the scene is required.

Spectral mixture analysis techniques are improved by a per pixel unmixing strategy. Stepwise regression provides a mathematical framework with which one adjusts the number of endmembers in the model to reduce overfit. Error checking of the fractions, and entrance and exit criteria, result in an automated algorithm that provides superior material maps. The algorithm was demonstrated for various constraint conditions. The effect of texture was examined as well. Mismatches between the spectral library and the image data are inevitable. Yet, with proper choice of the library, reasonable unmixing can be achieved.

Finally, the spectral mixing methodology was extended into the image fusion domain by a process called sharpening. The model for sharpening takes the same form as the spectral mixing model, but in the high resolution case, the problem is an under-determined optimization. The results show

sharpening material maps yields more accurate fractions than if the maps are not sharpened. Thus, fusing high resolution data with the spectral mixture maps results in a better description of the underlying scene.

Sharpening was demonstrated at multiple scales. Flexibility in combining images of different scales is important because future fusion applications will be varied. One desires an algorithm that works with different imaging sensors.

Sharpening can be done with one or more sharpening images. The multiple band capability is especially significant. The results demonstrate that selection of the sharpening band affects performance. One should choose which bands to apply based upon which materials are of most interest. Although visible band fusion is likely the first application, the additional information contained in the NIR and SWIR regions should not be neglected. There is compelling evidence to add high spatial resolution infrared bands to future systems for the expressed purpose of sharpening other spectral sensors. The payoff from fusing with a high resolution infrared band may be even more than the improvement gained from fusing with a high resolution visible image. It also seems clear that fusing with both sets of data would give an even better result.

6.2 Limitations

Several limitations to these results must be noted. The most obvious is that the quantitative data were shown for a single test scene. Prior fusion studies, as well as the texture vs. no texture results, indicate that the algorithm performance is affected by variation in the image. Therefore, one should study multiple images, with different amounts of materials and textures, to determine the robustness of the algorithm.

Another limitation of these results in the broad classes of materials in the spectral library. The data were only generated for a 15 band sensor. Distinguishing vehicles from grass is not especially difficult. While separating the grass and trees was slightly more challenging, the true value of hyperspectral data is separating different groups within the same material class (e.g., types of trees, etc.). This analysis did not attempt such detailed classification, nor was its performance compared to digital count based classification techniques.

94

Finally, one of the fundamental assumptions during this work was that the images were accurately registered. Misalignments in the images, as well as distortions due to acquisition geometry will degrade the results. This work also did not address the effects of sensor calibration errors or atmospheric effects.

6.3 Recommendations

The limitations discussed above could be addressed by testing the algorithm on different images and with different types of materials. One may need to synthetically generate hyperspectral images. This would require detailed data bases which capture the spectral character of the materials in order to obtain realistic results. One could study algorithm performance as a function of the amount of variation and distortion. As part of the study, one could include sensor calibration and atmospheric correction to determine their effects on the results. The more error sources that exist, the more the measurements will vary from the predictions in the library. This, in turn, results in errors in the fraction maps.

Another area of investigation is the order of unmixing and sharpening. For example, if multiple sharpening bands are available, one could classify first, then unmix (Zhukov, et al., 1996). Alternately, one could use the fusion techniques discussed in section two, and then unmix the high resolution multispectral images. Finally, one could remix the material maps to return to the digital count domain. This could be used to predict, or synthesize, how other high resolution spectral bands might appear. This may or may not be useful, and is likely to be inaccurate, as unmixing errors would be applied twice.

One could test the performance using in-scene derived endmembers. While this removes the requirement of a spectral library, equating the scene endmembers with useful material classes is more difficult. In addition, because we desire to fuse the spectral sensor with the spatial sensor, the same materials must be identifiable at both resolutions. If the low resolution endmembers are not pure, it may be difficult to find corresponding high resolution pixels containing the exact same proportions of the materials. It is unlikely in-scene endmembers can be extracted at high resolution. If this were the case, one would not need the spectral data. If one desires to use scene derived endmembers, the sharpening library will have to be formed by integrating the spectral library over the band passes of the sharpening bands.

Another area of research is to incorporate the statistics that accompany the regression analysis. For example, as part of the least squares calculations, one develops an estimate of the variance. From this, one could calculate confidence bounds on the model. This could be used to influence model selection, or as an error checking mechanism to identify poor unmixing. A useful, automated algorithm must handle a large percentage of the pixels, but marking bad data may be an acceptable compromise between fast performance, accurate results, and flexibility in searching for unexpected materials. One might also attempt to model the library errors. Stepwise regression assumes the errors are gaussian and independent. One might characterize the mismatch between the library and the image data, and investigate that effect on the stepwise based unmixing.

Along this line, it may be fruitful to orthogonalize the process. If a principle components transformation were applied to the spectral library, the resulting endmembers would be orthogonal. An identical transformation is then applied to the image digital counts (reflectances). Once the data are transformed, the errors in the regression analysis may be better behaved, making the stepwise unmixing algorithm more accurate. Of course, the cost of doing this transformation is increased computations and a loss of correspondence between the endmembers and the materials in the scene.

A list of recommendations would be incomplete if it did not include a suggestion to recode the algorithm. The pixel processing in Matlab was accomplished without the Image Processing Toolbox. A large amount of coding overhead was required to manipulate the images into vectors and two dimensional matrices. Images are better represented as cubes to account for their spectral nature. A different software package is recommended in this regard. In addition, little effort was expending making the code efficient. For example, it is possible to implement a very effective stepwise regression algorithm using partial correlation coefficients (Draper and Smith, 1966). This specialized algorithm is incompatible with unmixing constraints, but this research concluded the best way to incorporate the constraints is after the endmembers are determined. Future implementations should use the efficient algorithm.

Finally, it seems clear that the entire image fusion approach should be influenced by more than the data within the superpixel. One should use a *spatial strategy* that incorporates knowledge of the

materials in surrounding pixels into the estimates for the fractions. The materials have a spatial correlation that is not accounted for by current techniques.

A spatial strategy could be hierarchical – identifying the pixels with accurate models (small RMS error) first. Then, knowing the materials in those pixels with confidence, one could probabilistically influence the unmixing of neighboring pixels.

The spatial strategy could be used to overcome some of the effects of variation. For example, without knowledge of the neighboring materials, one would unmix based upon the digital counts and the spectral library. Mismatches result in poor fit. However, if one knew surrounding materials contained a particular endmember, a spatial strategy might "help" that material into the model. The unmixing algorithm could be designed to require a particular material be used in the model, or it could weight the regression to simply make it more likely that material would get used. Alternately, one could structure the problem in a probabilistic manner, so that the material fractions represent the a posteriori probability of a material being present, subject to evidence provided by the digital counts, library, and surrounding pixels.

In conclusion, this image fusion algorithm applies spectral mixing models at both low and high resolutions in order to construct high resolution material maps. The unmixing and sharpening steps are solved as linear least squares problems. The algorithm works at a variety of scales and constraint conditions, with multiple sharpening bands, and in the presence of image texture. The results show a quantifiable improvement over traditional unmixing techniques. Future investigation into the strengths and weaknesses of this approach is warranted.

7. Appendices

Appendix A: Analytical Solution to Equality Constrained Over Determined Least Squares Problem

An over determined least squares problem with linear equality constraints is defined as

$$\min F(x) = (y - \hat{y})^2 = (y - Ax)^2 = y'y - 2y'Ax + x'A'Ax$$
$$\text{subject to } Cx = d \tag{A-1}$$

where y is an m-vector, x is an n-vector, d is a p-vector, A is $m \times n$, C is $p \times n$, and $m > n > p$. The problem can be written using Lagrange multipliers in the following form

$$\begin{bmatrix} A'A & C' \\ C & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} A'y \\ d \end{bmatrix}. \tag{A-2}$$

An analytical solution to the matrix inverse is

$$\begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} (A'A)^{-1} - WC(A'A)^{-1} & W \\ \left[C(A'A)^{-1}C' \right]^{-1} C(A'A)^{-1} & -\left[C(A'A)^{-1}C' \right]^{-1} \end{bmatrix} \begin{bmatrix} A'y \\ d \end{bmatrix} \tag{A-3}$$

where

$$W = (A'A)^{-1}C' \left[C(A'A)^{-1}C' \right]^{-1}. \tag{A-4}$$

One can verify the inverse is correct.

$$\begin{bmatrix} (A'A)^{-1} - WC(A'A)^{-1} & W \\ \left[C(A'A)^{-1}C' \right]^{-1} C(A'A)^{-1} & -\left[C(A'A)^{-1}C' \right]^{-1} \end{bmatrix} \begin{bmatrix} A'A & C' \\ C & 0 \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ 0 & I_p \end{bmatrix} \tag{A-5}$$

Therefore, a closed form solution for x is

$$x = x_u + W(d - Cx_u) \tag{A-6}$$

where $x_u = (A'A)^{-1}A'y$ is the usual unrestricted estimator (Draper and Smith, 1981).

Appendix B: Gradient Projection Algorithm

The gradient projection algorithm was used in an earlier implementation of fully constrained sharpening. It provides an intuitive insight into the optimization problem. The negative gradient of an unconstrained objective function points in the direction of *steepest descent*. Stepping in the direction of the negative gradient is a popular way to find a local minimum of a function.

Define

$$e(x) = y - \hat{y} = y - Ax.$$

(B-1)

A new value of x that reduces the cost function could be found by moving in the direction of the negative gradient of x, e.g.,

$$x_{k+1} = x_k - \alpha \nabla e(x_k)$$

(B-2)

where α is a scalar step size, and ∇ is the gradient operator. If there are constraints at x_k, (either via equality constraints or active inequality constraints), the allowable step direction is restricted. The *projected gradient* is this restricted allowable direction found by projecting the gradient onto the subspace corresponding to the orthogonal complement of the constraints.

Figure 43 illustrates the gradient projection. The subspace $\{M\}$ is the space spanned by all the active constraints at x_k.
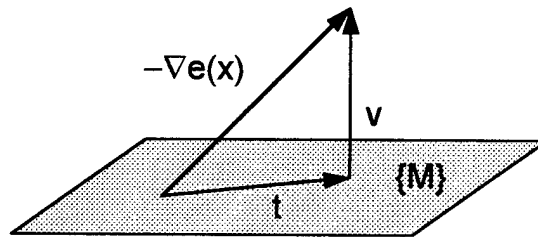
Figure 43: Projected Gradient

The negative gradient can be written in terms of two vectors, t_k contained in $\{M\}$ and v_k orthogonal to $\{M\}$,

$$-\nabla e(x_k) = t_k + v_k.$$

(B-3)

Since t_k must remain fixed to satisfy the constraints, the vector v_k is the gradient projection desired. If one steps in the direction of v_k, all constraints remain satisfied, and the cost function is reduced. Linear algebra is used to solve for v_k.

First, t_k is written as a linear combination of the q active constraints μ_i

$$t_k = \sum_{i=1}^{q} \lambda_i \mu_i' = M_k' \lambda,$$
(B-4)

where M_k is a $q \times n$ matrix, whose rows (μ_i) correspond to active constraints at x_k, λ is a q-vector of Lagrange multipliers, and t_k is the n-vector projection of $-\nabla e(x_k)$ into the q-dimensional row subspace of M_k. Note the active constraints are assumed to be non-degenerate, i.e. the q rows are linearly independent and $q \le n$. By definition, v_k is orthogonal to t_k, so

$$M_k v_k = 0.$$
(B-5)

Substituting (B-4) and (B-5) into (B-3) and premultiplying by M_k gives

$$- M_k \nabla e(x_k) = M_k M_k' \lambda.$$
(B-6)

Since rank $(M_k) = q$, the matrix $M_k M_k'$ is invertible and one can solve for the multipliers

$$\lambda = -(M_k M_k')^{-1} M_k \nabla e(x_k).$$
(B-7)

Now, use (B-3) to solve for v_k, substituting the results from (B-4) and (B-7),

$$\begin{aligned}
v_k &= -\nabla e(x_k) - t_k \\
&= -\left[\nabla e(x_k) + M_k' \lambda \right] \\
&= -\left[\nabla e(x_k) - M_k' (M_k M_k')^{-1} M_k \nabla e(x_k) \right] \\
&= -\left[I - M_k' (M_k M_k')^{-1} M_k \right] \nabla e(x_k) \\
&= -P \nabla e(x_k)
\end{aligned}$$
(B-8)

where

$$P = \left[I - M_k' (M_k M_k')^{-1} M_k \right]$$
(B-9)

is a projection matrix written in terms of the active constraints M_k, and I is an n x n identity matrix. Moving in the direction of the projected gradient reduces the cost function while satisfying all the active constraints. The algorithm steps in this direction until the function is minimized or another constraint becomes active. The Kuhn-Tucker conditions are used to determine convergence.

Recall for the least squares problem, the function to be minimized is quadratic.

$$
\begin{aligned}
F(x) &= \left(y - \hat{y}\right)^2 \\
&= \left(y - Ax\right)^2 \\
&= y'y - 2y'Ax + x'A'Ax
\end{aligned}
\tag{B-10}
$$

The quadratic term $Q = A'A$ is a symmetric n x n matrix. If rank $(A) = n < m$, as in the over-determined LS problem (unmixing), Q is positive definite and the local solution satisfies the sufficient conditions. In the under-determined case (sharpening), rank $(A) = m < n$, Q is positive semidefinite and only satisfies the necessary conditions. Zero eigenvalues in Q correspond to "flat" dimensions, which could trap the search performed by iterative algorithms like the gradient projection. Therefore, the semidefinite quadratic cost term is modified slightly to ensure positive eigenvalues. Also note that since the cost function is quadratic, the gradient projection will not trap the solution in the wrong minimum. With a quadratic cost function, a local minimum is also the global minimum.

The quadratic term Q can be made positive definite by using

$$
\widetilde{Q} = Q + \delta I(n)
\tag{B-11}
$$

where $I(n)$ is an n x n identity matrix and δ is a small positive constant The new matrix is positive definite since

$$
x'\widetilde{Q}x = x'Qx + \delta x'Ix = x'Qx + \delta \sum_{j=1}^{n} x_j^2,
\tag{B-12}
$$

and for $x \neq 0$, $x'\widetilde{Q}x > x'Qx \geq 0$. For a small δ, "the resulting change in Q would be so slight as to have only a negligible effect on the location of the optimal solution," (Simmons, 1975, pg. 236).

The gradient projection method was originally implemented because of its intuitive nature. The fully constrained solution is found by following the projected negative gradient along the active constraint

boundary until the constrained minimum is found. For the partially constrained condition, all constraints are active, and the algorithm proceeds directly to the solution.

The gradient projection algorithm requires identification of an initial feasible point. In this case, set the initial high resolution fractions equal to the corresponding low resolution fractions, i.e.

$$f^1 = f^2 = \cdots = f^s = f$$

$$x_1 = \begin{bmatrix} f \\ \hline f \\ \hline \vdots \\ \hline f \end{bmatrix} \tag{B-13}$$

If the high resolution measurements are equal, corresponding to no high resolution spatial information, the gradient projection returns the starting vector as desired. For this case (an aggregate mixture), the high resolution fractions should equal the low resolution fractions. If there is spatial information contained in the spatial sensor data, the sharpening algorithm adjusts the high resolution fractions to return the minimum length vector with the least square residual error.

The gradient projection was eventually replaced by the linear algebra technique based upon orthogonal decomposition and the Non Negative Least Squares algorithm. Although some insight was lost, the orthogonal decomposition method had improved numerical stability.

Appendix C: Spectral Libraries

M-7 Spectral Bands

The M-7 sensor has 15 bands in the visible through SWIR spectral regions. The lower and upper wavelengths for the band passes used by the DIRSIG algorithm are listed in Table 5. The last column in the table reflects the width of the spectral band.

Band Number	Low	High	Width
1	0.44	0.5	0.06
2	0.46	0.53	0.07
3	0.495	0.57	0.075
4	0.46	0.62	0.16
5	0.58	0.675	0.095
6	0.615	0.72	0.105
7	0.66	0.765	0.105
8	0.705	0.93	0.225
9	0.76	1.045	0.285
10	0.9	1.385	0.485
11	1.1	1.39	0.29
12	1.3	1.79	0.49
13	1.4	1.89	0.49
14	1.9	2.39	0.49
15	1.9	2.49	0.59

Table 5:   M-7 Spectral Bands (μm)

The specific reflectance values for the individual materials were calculated by averaging the DIRSIG material emissivity file over the bandpass (Table 6). The files contain anywhere from a single curve to several dozen realizations of the particular materials. All curves were averaged to create the basic library. The average emissivity was subtracted from one to get an average reflectance. The averaging algorithm was different than the one used by the DIRSIG code. This causes some mismatch between the library and the image values. In addition, the spectral library contains no angular effects.

Material	Camo. Paint	Water	Grass	Concrete	Rubber	Dirt	Wood	Trees	Painted Steel	Canvas
Band 1	0.1312	0.0713	0.0274	0.1543	0.0447	0.1487	0.2868	0.0636	0.1880	0.0410
Band 2	0.1362	0.0748	0.0350	0.1639	0.0435	0.1671	0.3179	0.0895	0.1801	0.0441
Band 3	0.1470	0.0793	0.0564	0.1809	0.0416	0.2048	0.3921	0.1382	0.1689	0.0488
Band 4	0.1559	0.0809	0.0659	0.1977	0.0399	0.2446	0.4929	0.1380	0.1625	0.0534
Band 5	0.1600	0.0739	0.0580	0.2113	0.0382	0.2827	0.6156	0.0930	0.1605	0.0578
Band 6	0.1609	0.0642	0.0721	0.2164	0.0371	0.3030	0.6933	0.1223	0.1621	0.0588
Band 7	0.1625	0.0519	0.1607	0.2210	0.0363	0.3265	0.7661	0.2108	0.1670	0.0594
Band 8	0.1606	0.0312	0.3236	0.2288	0.0360	0.3662	0.8459	0.3039	0.1876	0.0629
Band 9	0.1611	0.0186	0.3795	0.2347	0.0354	0.3923	0.8762	0.3096	0.2068	0.0659
Band 10	0.1452	0.0030	0.4115	0.2554	0.0333	0.4648	0.8689	0.2837	0.2634	0.0721
Band 11	0.1348	0.0005	0.4019	0.2684	0.0323	0.5026	0.8397	0.2604	0.2955	0.0747
Band 12	0.1310	0.0018	0.2600	0.2992	0.0364	0.5412	0.6684	0.1747	0.3727	0.0791
Band 13	0.1429	0.0009	0.2876	0.3099	0.0380	0.5922	0.6089	0.1583	0.3990	0.0798
Band 14	0.1306	0.0017	0.2413	0.3100	0.0366	0.5925	0.4198	0.0776	0.4970	0.0897
Band 15	0.1592	0.0014	0.3037	0.3057	0.0377	0.6203	0.4035	0.0707	0.5039	0.0898

Table 6: M-7 Spectral Bands (μm)

Figure 44 is a plot of the ten spectral reflectance curves used in the unmixing library.
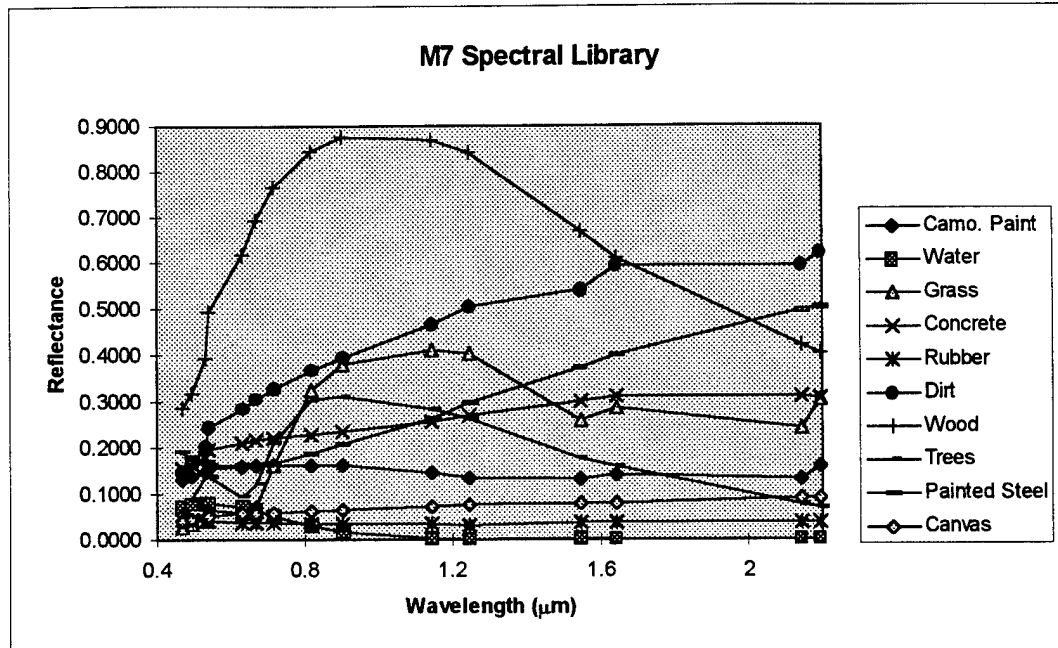
Figure 44: M-7 Spectral Reflectance Curves

104

When the New Library was calculated, the grass and tree library values were extracted from a single curve rather than from averaging all the curves in the data file. The old and new reflectance values are shown in Table 7.

Band	Grass: Averaged	Grass: Single	Trees: Averaged	Trees: Single
1	0.0274	0.0479	0.0636	0.0702
2	0.0350	0.0574	0.0895	0.1013
3	0.0564	0.0822	0.1382	0.1622
4	0.0659	0.0964	0.1380	0.1629
5	0.0580	0.0960	0.0930	0.1074
6	0.0721	0.1114	0.1223	0.1431
7	0.1607	0.1914	0.2108	0.2560
8	0.3236	0.3373	0.3039	0.3788
9	0.3795	0.3883	0.3096	0.3867
10	0.4115	0.4215	0.2837	0.3560
11	0.4019	0.4162	0.2604	0.3282
12	0.2600	0.3053	0.1747	0.2197
13	0.2876	0.3423	0.1583	0.1979
14	0.2413	0.3029	0.0776	0.0957
15	0.3037	0.3568	0.0707	0.0869

Table 7: New Spectral Library Reflectance Values for Grass and Trees

Sharpening Bands

The synthetic M-7 data was also used to create sharpening bands. The visible sharpening band was a weighted average of M-7 bands 4 and 6. Band 9 (NIR) was used as the second sharpening band, and band 13 (SWIR) was used as sharpening band 3. The sharpening bands are summarized in Table 8.

Sharpening Band Number	Derived From	Low	High	Width
1	$\dfrac{0.145*Band4 + 0.105*Band6}{0.26}$	0.46	0.72	0.26
2	M-7 Band 9	0.76	1.045	0.285
3	M-7 Band 13	1.4	1.89	0.49

Table 8: Sharpening Bands (μm)

The specific reflectance values for the individual materials in the sharpening bands are listed in Table 9.

Material	Camo. Paint	Water	Grass	Concrete	Rubber	Dirt	Wood	Trees	Painted Steel	Canvas
Band 1	0.1579	0.0742	0.0684	0.2052	0.0388	0.2682	0.5738	0.1316	0.1623	0.0556
Band 2	0.1611	0.0186	0.3795	0.2347	0.0354	0.3923	0.8762	0.3096	0.2068	0.0659
Band 3	0.1429	0.0009	0.2876	0.3099	0.0380	0.5922	0.6089	0.1583	0.3990	0.0798

Table 9: Reflectance Values for Sharpening Library

Figure 45 is a plot of the ten spectral reflectance curves used in the three sharpening bands.
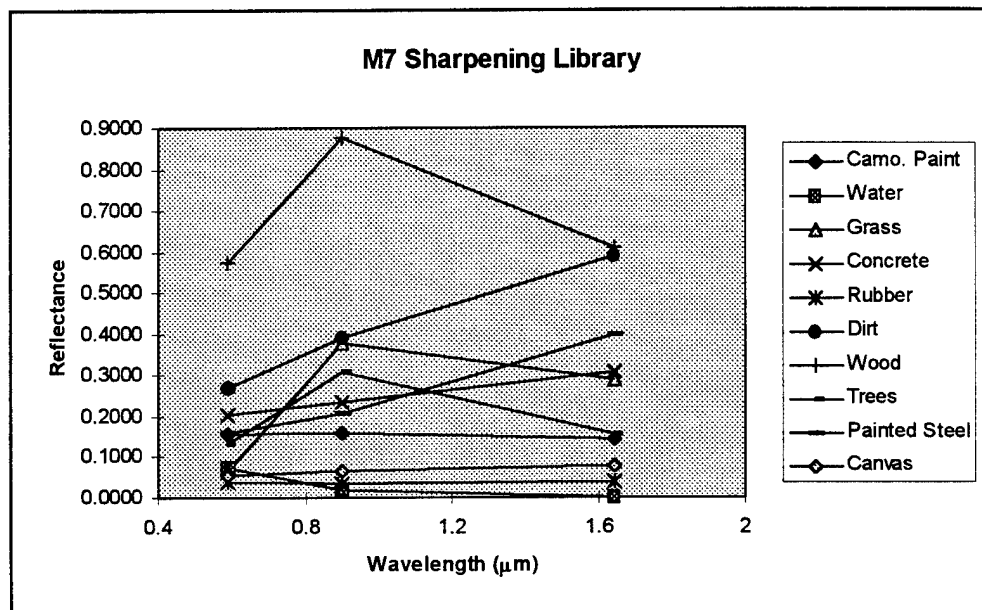
Figure 45: M-7 Sharpening Library

When the New Library was calculated, the grass and tree library values were extracted from a single curve rather than from averaging all the curves in the data file. The old and new reflectance values are shown in Table 10.

	Grass: Averaged	Grass: Single	Trees: Averaged	Trees: Single
Band 1	0.0684	0.1025	0.1316	0.1549
Band 2	0.3795	0.3883	0.3096	0.3867
Band 3	0.2876	0.3423	0.1583	0.1979

Table 10: New Sharpening Library Reflectance Values for Grass and Trees

Appendix D: Data Sets

Following are the data used to create the graphs in the results section (Table 11). The operations listed are replication, sharpening, and unmixing. The second column shows the scale of the original data set. The third column indicates whether the image contained texture variation. The fourth column identifies the source of the data. Perfect indicates perfectly unmixed maps. A number indicates the low resolution maps were unmixed with the corresponding constraints. The constraint key is 0 = unconstrained, 1 = partial constraints (option 2), 1.5 = partial constraints (option 1), 2 = full constraints (option 2), and 2.5 = full constraints (option 1). Recall option 1 indicates the constraints were included in the stepwise algorithm. For option 2, the unmixing was done unconstrained to determine the endmembers. Then the coefficients were constrained for that set of materials.

If sharpening, the final scale indicates the scale after sharpening was applied. If replicating or unmixing, the final scale identifies which maps the data were compared against (using replication if they are of different scales). The next column indicates the sharpening constraint condition. Finally, the sharpening bands are labeled. Each digit corresponds to a band. The total error is the sum of the squared error between the test and reference fractions. The error per pixel is found by dividing the total error by the number of pixels in the reference image.

Operation	Initial Scale	Other	Source	Final Scale	Constraint	Bands	Total Error	Error per Pixel
Replicate	4	Texture	Perfect	2			1714.3750	0.0496
Replicate	8	Texture	Perfect	2			4092.3594	0.1184
Replicate	16	Texture	Perfect	2			6366.0000	0.1842
Replicate	32	Texture	Perfect	2			8208.7183	0.2375
Replicate	8	Texture	Perfect	4			594.4961	0.0688
Replicate	16	Texture	Perfect	4			1162.9063	0.1346
Replicate	32	Texture	Perfect	4			1623.5858	0.1879
Replicate	16	Texture	Perfect	8			142.1025	0.0658
Replicate	32	Texture	Perfect	8			257.2724	0.1191
Replicate	32	Texture	Perfect	16			28.7925	0.0533
Replicate	16	No Texture	Perfect	4			878.6772	0.1017
Sharpen	8	Texture	Perfect	4	0	1	603.4512	0.0698
Sharpen	8	Texture	Perfect	4	1	1	480.9644	0.0557
Sharpen	8	Texture	Perfect	4	2	1	425.4226	0.0492
Sharpen	16	Texture	0	4	0	1	2194.6951	0.2540
Sharpen	16	Texture	1	4	1	1	2055.9532	0.2380

Sharpen	16	Texture	1.5	4	1	1	8295.0125	0.9601
Sharpen	16	Texture	2	4	2	1	2040.7113	0.2362
Sharpen	16	Texture	2.5	4	2	1	9699.8555	1.1227
Sharpen	16	Texture	2	4	2	13	1518.9659	0.1758
Sharpen	16	Texture	2	4	2	123	1494.9305	0.1730
Sharpen	16	Texture	Perfect	4	0	1	1129.6315	0.1307
Sharpen	16	Texture	Perfect	4	1	1	932.0370	0.1079
Sharpen	16	Texture	Perfect	4	2	1	906.0598	0.1049
Sharpen	16	Texture	Perfect	4	0	2	1164.9063	0.1348
Sharpen	16	Texture	Perfect	4	1	2	1084.2013	0.1255
Sharpen	16	Texture	Perfect	4	2	2	878.0093	0.1016
Sharpen	16	Texture	Perfect	4	0	3	1124.8589	0.1302
Sharpen	16	Texture	Perfect	4	1	3	911.1458	0.1055
Sharpen	16	Texture	Perfect	4	2	3	341.7203	0.0396
Sharpen	16	Texture	Perfect	4	1	12	598.5369	0.0693
Sharpen	16	Texture	Perfect	4	2	12	444.8245	0.0515
Sharpen	16	Texture	Perfect	4	1	13	594.0799	0.0688
Sharpen	16	Texture	Perfect	4	2	13	127.0558	0.0147
Sharpen	16	Texture	Perfect	4	1	23	903.4808	0.1046
Sharpen	16	Texture	Perfect	4	2	23	205.5187	0.0238
Sharpen	16	Texture	Perfect	4	1	123	611.9409	0.0708
Sharpen	16	Texture	Perfect	4	2	123	121.8358	0.0141
Sharpen	16	No Texture	2	4	2	1	1353.2855	0.1566
Sharpen	16	No Texture	Perfect	4	0	1	809.6493	0.0937
Sharpen	16	No Texture	Perfect	4	1	1	112.5155	0.0130
Sharpen	16	No Texture	Perfect	4	2	1	139.3016	0.0161
Sharpen	16	New Lib	2	4	2	1	618.4432	0.0716
Sharpen	16	New Lib	2	4	2	123	150.8533	0.0175
Sharpen	16	New Lib	Perfect	4	0	1	846.6139	0.0980
Sharpen	16	New Lib	Perfect	4	1	1	74.5216	0.0086
Sharpen	16	New Lib	Perfect	4	2	1	76.6497	0.0089
Sharpen	16	New Lib	Perfect	4	2	123	25.0240	0.0029
Sharpen	32	Texture	Perfect	4	0	1	1560.3238	0.1806
Sharpen	32	Texture	Perfect	4	1	1	1314.8908	0.1522
Sharpen	32	Texture	Perfect	4	2	1	1249.7535	0.1446
Unmix	4	Texture	Perfect	4	0		1454.8671	0.1684
Unmix	4	Texture	Perfect	4	1.5		9988.9739	1.1561
Unmix	4	Texture	Perfect	4	1		1078.8170	0.1249
Unmix	4	Texture	Perfect	4	2.5		11233.8210	1.3002
Unmix	4	Texture	Perfect	4	2		1314.7049	0.1522
Unmix	8	Texture	Perfect	8	0		280.4302	0.1298
Unmix	8	Texture	Perfect	8	1.5		1971.3977	0.9127
Unmix	8	Texture	Perfect	8	1		295.3665	0.1367
Unmix	8	Texture	Perfect	8	2.5		2329.6188	1.0785
Unmix	8	Texture	Perfect	8	2		289.8102	0.1342
Unmix	16	Texture	Perfect	4	0		2251.7741	0.2606
Unmix	16	Texture	Perfect	4	1.5		8308.7204	0.9617
Unmix	16	Texture	Perfect	4	1		2269.9791	0.2627

Unmix	16	Texture	Perfect	4	2.5		9621.0801	1.1136
Unmix	16	Texture	Perfect	4	2		2262.8620	0.2619
Unmix	16	Texture	Perfect	16	0		68.0542	0.1260
Unmix	16	Texture	Perfect	16	1.5		446.6134	0.8271
Unmix	16	Texture	Perfect	16	1		69.1921	0.1281
Unmix	16	Texture	Perfect	16	2.5		528.6359	0.9790
Unmix	16	Texture	Perfect	16	2		68.7472	0.1273
Unmix	16	No Texture	No Dirt	16	0		41.6457	0.0771
Unmix	16	No Texture	No Dirt	16	1.5		916.4403	1.6971
Unmix	16	No Texture	No Dirt	16	1		67.1457	0.1243
Unmix	16	No Texture	No Dirt	16	2.5		903.5486	1.6732
Unmix	16	No Texture	No Dirt	16	2		32.9764	0.0611
Unmix	16	No Texture	Perfect	16	0		36.5210	0.0676
Unmix	16	No Texture	Perfect	16	1.5		914.2547	1.6931
Unmix	16	No Texture	Perfect	16	1		48.8038	0.0904
Unmix	16	No Texture	Perfect	16	2.5		903.3162	1.6728
Unmix	16	No Texture	Perfect	16	2		36.0492	0.0668
Unmix	16	New Lib	Perfect	16	0		9.6313	0.0178
Unmix	16	New Lib	Perfect	16	1		4.4894	0.0083
Unmix	16	New Lib	Perfect	16	2		3.8563	0.0071
Unmix	16	New Lib	Perfect	4	2		940.3787	0.1088

Table 11: Detailed Results

Appendix E: Statistical Significance of Fusion Results

Hypothesis tests are used to determine if using the proposed algorithm has a significant effect on the results. The error metric used to compare algorithms is a normalized sum of the squared differences between reference fractions and test fractions.

$$SE = \frac{1}{N} \sum_{pixels} \sum_{materials} \left(f_{truth} - f_{test} \right)^2$$ (E-1)

The inner sum is the amount of squared error in each pixel. That error is then summed and normalized (averaged) over the image. If the spatial distribution of the error is not significant, one could analyze the N-vector of pixel errors. Since the pixel errors themselves consist of the sum of squared material errors, the error distribution will not be gaussian. Figure 46 compares a representative error distribution with a gaussian distribution having the same mean and variance.
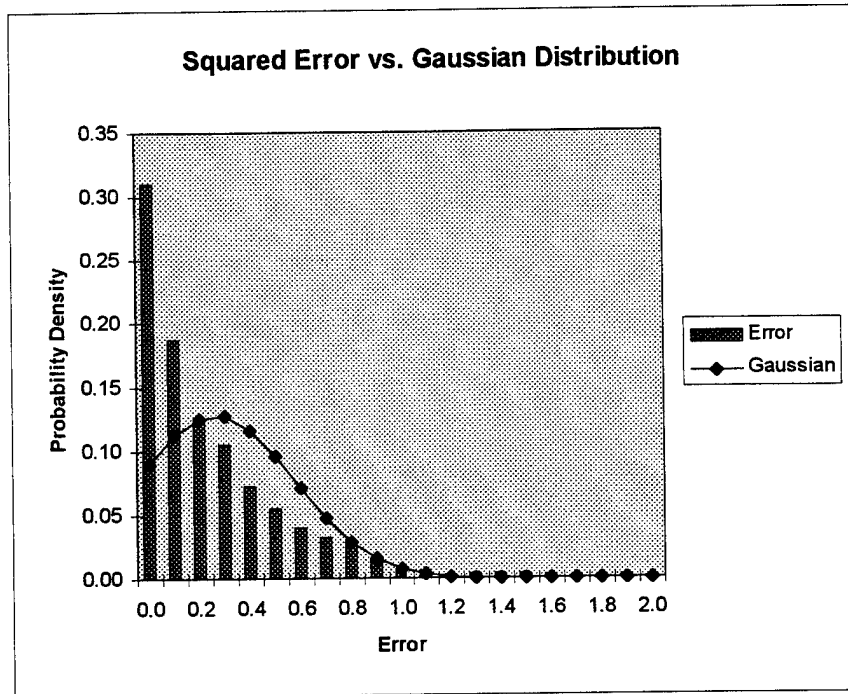

Figure 46: Squared Error vs. Gaussian Distribution

111

A goodness of fit test confirms the errors are not close enough to a normal distribution. However, the number of pixels (data points) in the image is large. Therefore, although the errors are not gaussian, they are tending towards gaussian due to the central limit theorem.

The error metric compares material maps constructed with a particular algorithm to a reference (truth) set of maps. If maps from two different algorithms are compared to the same truth data, we can examine the two error metrics to detect a difference. In fact, the normalized error metric is the mean of the N-vector of pixel squared errors. One desires to test the significance of the difference between two means (Dougherty, 1990). In other words, is the improvement (reduction in the mean) of one method over another due to the algorithm or to random chance. Table 12 contains the sample mean and variance derived from the data sets plotted in Figure 35: Fusion with a Single Sharpening Band. There were $N = 72*120 = 8640$ elements (pixels) in each data set.

Data Set	Sample Mean	Sample Variance
No Sharpening	0.2619	0.3127
Unconstrained Fusion	0.2540	0.3019
Partially Constrained Fusion	0.2380	0.3040
Fully Constrained Fusion	0.2362	0.3035

Table 12: Statistics from Single Band Fusion Data Sets

Let X correspond to the first N-vector of errors, and Y correspond to the second vector. Further, let Y correspond to the "improved" algorithm, so that we are testing whether the reduction in the mean of Y is reduced "enough" over the mean of X. Assume the elements of X and Y are independent of each other. Note this implies no spatial correlation in the pixel errors, and will not be true if the wrong endmembers are used. Suppose random variables X and Y have means μ_X and μ_Y and standard deviations σ_X and σ_Y. The null hypothesis is that the means are equal.

$$H_0: \ \mu_X - \mu_Y = 0 \qquad\qquad \text{(E-2)}$$

The alternate hypothesis is that the means are not equal (and $\mu_Y < \mu_X$).

$$H_1: \ \mu_X - \mu_Y > 0 \qquad\qquad \text{(E-3)}$$

Now, consider the difference between the sample means. Since the sample means are unbiased estimators of the true means, and X and Y are assumed independent, the test statistic, Z, has the following expected mean and variance

$$Z = \overline{X} - \overline{Y}$$
$$\mu_Z = \mu_{\overline{X} - \overline{Y}} = \mu_X - \mu_Y \qquad (E-4)$$
$$\sigma_Z^2 = \sigma_{\overline{X} - \overline{Y}}^2 = \frac{\sigma_X^2 + \sigma_Y^2}{N}$$

The variances of X and Y are estimated by the sample variances. We assume those estimates are good because of the large sample size. If one knows X and Y are normally distributed, then Z is also normal. For the data in Figure 35 (squared errors), X and Y are *not* normally distributed, but Z may or may not be normal. Two hypothesis tests are demonstrated. In the first, Z is assumed to have a normal distribution. The second test is called a rank-sum test, and requires no distributional assumptions.

If Z is gaussian, the ratio of its mean to its standard deviation can be compared to a table of "standard normal" values to determine the likelihood (significance) of the difference in the means. If the null hypothesis were true, then $Z = 0$. If $\mu_X > \mu_Y$, $Z > 0$. The standard normal table gives the probability of that deviation occurring. One then decides whether to accept or reject the null hypothesis.

Alternatively, the rank-sum test can be used without making distributional assumptions. One constructs a $2*N$-vector by placing X and Y into a single vector. The elements of this vector are ranked (sorted) in increasing order. The test statistic, W, is formed by summing all the ranks corresponding to X. Under the null hypothesis, W has an expected mean and variance

$$\mu_W = \frac{N(2N+1)}{2}$$
$$\sigma_W^2 = \frac{N^2(2N+1)}{12} \qquad (E-5)$$

Obviously, if X and Y are identically distributed, $W = \mu_W$. If every element of X is smaller than every element of Y, W is a minimum. Similarly, if every element of X is larger than every element of Y, W will be a maximum. With the large number of samples in this example, W is assumed to be normally

distributed. One then relates the deviation in the rank-sum, $Z_W = (W-\mu_W)/\sigma_W$, to a standard normal distribution.

Table 13 contains the test statistics Z and Z_W for selected pairs of data from Figure 35. The probability of a gaussian random variable having a value of the test statistic under the null hypothesis is shown. In the table, low probabilities indicate the test statistic is unlikely to occur under H_0. If the probability satisfies a required confidence level, one rejects the null hypothesis in favor of concluding the algorithm caused an improvement.

Data Sets	Z	probability $(Z \mid H_0)$	W	Z_W	probability $(Z_W \mid H_0)$
No Sharpening vs. Unconstrained	1.69	0.0455	74746719	0.28	0.3897
No Sharpening vs. Partial Constraints	5.09	< 0.0001	76165940	4.61	< 0.0001
No Sharpening vs. Full Constraints	5.48	< 0.0001	76872981	6.77	< 0.0001
Partial vs. Full Constraints	0.38	0.3520	75381604	2.22	0.0132

Table 13: Test Statistics for Single Band Fusion Results

The table shows the improvement of unconstrained fusion over no sharpening is slight. If Z were gaussian, we could reject the null hypothesis with 95% confidence. However, the rank-sum test indicates a much greater chance the improvement was due to random variation. Therefore, it is unclear whether it is legitimate to conclude unconstrained fusion is better. Fortunately, the constrained fusion improvements are significant using both tests to more than 99.9% confidence. The last case also shows a disparity in the statistics. When comparing partial vs. fully constrained fusion, the gaussian assumption indicates the difference is not significant. However, the rank-sum test indicates the null hypothesis could be rejected with 98% confidence.

References

1. Adams, J.B., Smith, M.O., Gillespie, A.R., "Imaging Spectroscopy: Interpretation Based on Spectral Mixture Analysis," *Remote Geochemical Analysis: Elemental and Mineralogical Composition*, Ed. C.M. Pieters and P.A.J. Englert, 145-166, 1993.

2. Adams, J.B., Smith, M.O., Johnson, P.E., "Spectral Mixture Modeling: A New Analysis of Rock and Soil Types at the Viking Lander 1 Site," *Journal of Geophysical Research*, 91, B8, 8098-8112, 1986.

3. Boardman, J.W., Kruse, F.A., Green, R.O., "Mapping Target Signatures Via Partial Unmixing of AVIRIS Data," *Summaries of the Fifth Annual JPL Airborne Earth Science Workshop, January 23-26, 1995*, Volume 1. AVIRIS Workshop, 23-26, 1995.

4. Borel, C.C., Gerstl, S.A.W., "Nonlinear Spectral Mixing Models for Vegetative and Soil Surfaces," *Remote Sensing of Environment*, 47, 403-416, 1994.

5. Borel, C.C., Gerstl, S.A.W., Powers, B.J., "The Radiosity Method in Optical Remote Sensing of Structured 3-D Surfaces," *Remote Sensing of Environment*, 36, 13-44, 1991.

6. Braun, G., "Quantitative Evaluation of Six Multi-spectral, Multi-resolution Image Merger Routines," *Master's Thesis*, RIT, 1992.

7. Bryson, Jr., A.E., Ho, Y., *Applied Optimal Control*, 1969.

8. Carper, W.J., Lillesand, T.M., Kiefer, R.W., "The Use of Intensity-Hue-Saturation Transformations for Merging SPOT Panchromatic and Multispectral Image Data," *Photogrammetric Engineering and Remote Sensing*, 56, 4 (April), 459-467, 1990.

9. Chavez, P.S., Jr., Sides, S.C., Anderson, J.A., "Comparison of Three Different Methods to Merge Multiresolution and Multispectral Data: Landsat TM and SPOT Panchromatic," *Photogrammetric Engineering and Remote Sensing*, 57, 3 (March), 295-303, 1991.

10. Crist, E.P., Cicone, R.C., "Application of the Tasseled Cap Concept to Simulated Thematic Mapper Data," *Photogrammetric Engineering and Remote Sensing*, 50, 3 (March), 343-352, 1984.

11. Dobson, A. J., *An Introduction to Generalized Linear Models*, 1990.

12. Dougherty, E.R., *Probability and Statistics for the Engineering, Computing, and Physical Sciences*, 626-632, 1990.

13. Dougherty, E.R., Personal Communication, 1996.

14. Draper, N.R., Smith, H., *Applied Regression Analysis*, 1st Ed., 1966.

15. Draper, N.R., Smith, H., *Applied Regression Analysis*, 2nd Ed., 1981.

16. Ehlers, M., "Multisensor Image Fusion Techniques in Remote Sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, 46, 19-30, 1991.

17. Evans, D.L., Smith, M.O., "Separation of Vegetation and Rock Signatures in Thematic Mapper and Polarimetric SAR Images," *Remote Sensing of Environment*, 37, 63-75, 1991.

18. Farrand, W.H., Singer, R.B., Merenyi, E., "Retrieval of Apparent Surface Reflectance from AVIRIS Data: A Comparison of Empirical Line, Radiative Transfer, and Spectral Mixture Methods," *Remote Sensing of Environment*, 47, 311-321, 1994.

19. Filberti, D.P., Marsh, S.E., Schowengerdt, R.A., "Synthesis of Imagery With High Spatial and Spectral Resolution From Multiple Image Sources," *Optical Engineering*, 33, 8 (August), 2520-2528, 1994.

20. Foley, T.M., "Zooming in on Remote Sensing Markets," *Aerospace America*, October 1994, 22-27, 1994.

21. Gao, B., Heidebrecht, K.B., Goetz, A,F,H,, "Derivation of Scaled Surface Reflectances from AVIRIS Data," *Remote Sensing of Environment*, 44, 165-178, 1993.

22. Gelb, A., *Applied Optimal Estimation*, 1974.

23. Gillespie, A.R., "Spectral Mixture Analysis of Multispectral Thermal Infrared Images," *Remote Sensing of Environment*, 42, 137-145, 1992.

24. Green, R.O., Conel, J.E., Roberts, D.A., "Estimation of Aerosol Optical Depth, Pressure Elevation, Water Vapor and Calculation of Apparent Surface Reflectance from Radiance Measured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Using A Radiative Transfer Code," *Proceedings of SPIE: Imaging Spectrometry of the Terrestrial Environment*, 1937, 14-15 April 1993, 2-11, 1993.

25. Gross, H.N., Schott, J.R., "Application of Spectral Mixing to Image Fusion," *Proceedings of 26th International Symposium on Remote Sensing of Environment*, 25-29 March 1996, Vancouver B.C., 428-431, 1996.

26. Gross, H.N., Schott, J.R., "Evaluating an Image Fusion Algorithm with Synthetic Image Generation Tools," *Proceedings of SPIE: Algorithms for Multispectral and Hyperspectral Imagery*, Vol. 2758, 8-12 April, Orlando FL, 1996.

27. Gross, H.N., Schott, J.R., "Application of Spatial Resolution Enhancement and Spectral Mixture Analysis to Hyperspectral Images," *Proceedings of SPIE: Hyperspectral Remote Sensing and Applications*, Vol. 2821, 4-9 August, Denver CO, 1996.

28. Iverson, A.E., Lersch, J.R., "Adaptive Image Sharpening Using Multiresolution Representations," *Proceedings of SPIE: Algorithms for Multispectral and Hyperspectral Imagery*, 2231, 5 April 1994, 72-83, 1994.

29. Johnson, H.K., Green, R.O., "AVIRIS User's Guide," *Summaries of Fifth Annual JPL Airborne Earth Science Workshop*, Ed. R.O. Green, Vol. 1., 23-26 Jan 1995, 105-108, 1995.

30. Kennedy, Jr., W.J., Gentle, J.E., *Statistical Computing*, 1980.

31. Lawson, C.L., Hanson, R.J., *Solving Least Squares Problems*, 1974.

32. Markham, B.L., Barker, J.L., "Spectral Characterization of the Landsat Thematic Mapper Sensors," *Landsat-4 Science Characterization Early Results: Volume II - Thematic Mapper (TM), Part I*, NASA Conf. Pub 2355, 235-276, 1985.

33. Mertes, L.A.K., Smith, M.O., Adams, J.B., "Estimating Suspended Sediment Concentrations in Surface Waters of the Amazon River Wetlands from Landsat Images," *Remote Sensing of Environment,* 43, 281-301, 1993.

34. Miller, A.J., *Subset Selection in Regression,* 1990.

35. MUG, "Important Spectral Regions for Representative Applications (Appendix II)," *Multispectral Users Guide,* Geodynamics Corp., August, 1995.

36. Munechika, C.K., "Merging Panchromatic and Multispectral Images for Enhanced Image Analysis," *Master's Thesis,* RIT, 1990.

37. Munechika, C.K., Warnick, J.S., Salvaggio, C., Schott, J.R., "Resolution Enhancement of Multispectral Image Data to Improve Classification Accuracy," *Photogrammetric Engineering and Remote Sensing,* 59, 1 (January), 67-72, 1993.

38. Mustard, J.F., "Relationships of Soil, Grass, and Bedrock over the Kaweah Serpentine Melange through Spectral Mixture Analysis of AVIRIS Data," *Remote Sensing of Environment,* 44, 293-308, 1993.

39. National Aeronautics and Space Administration, "MODIS: Moderate-Resolution Imaging Spectroradiometer," *1995 MTPE EOS Reference Handbook,* Goddard Space Flight Center, MD, 151-154, 1995.

40. Novo, E.M., Shimabukuro, Y.E., "Spectral Mixture Analysis of Inland Tropical Waters," *International Journal of Remote Sensing,* 15, 6, 1351-1356, 1994.

41. Pradines, D., "Improving Spot Images Size and Multispectral Resolution," *Proceedings of SPIE: Earth Remote Sensing Using the Landsat Thematic Mapper and SPOT Sensor Systems,* 660, 98-102, 1986.

42. Price, J.C., "How Unique Are Spectral Signatures?," *Remote Sensing of Environment,* 49, 181-186, 1994.

43. Price, J.C., "Combining Panchromatic and Multispectral Imagery from Dual Resolution Satellite Instruments," *Remote Sensing of Environment,* 21, 119-128, 1987.

44. Ranchin, T., Wald, L., "Merging SPOT-P and KVR-1000 Images for Updating Urban Maps," *Proceedings of 26th International Symposium on Remote Sensing of Environment,* 25-29 March 1996, 401-404, 1996.

45. Ranchin, T., Wald, L., Mangolini, M., "Efficient Data Fusion Using Wavelet Transform: The Case of SPOT Satellite Images," *Proceedings of SPIE: Wavelet Applications in Signal and Image Processing,* 2034, 15-16 July 1993, 171-178, 1993.

46. Research Systems, Inc., *ENVI: The Environment for Visualizing Images,* Version 2.0, July, 1995, Boulder, CO, 1995.

47. Richards, J.A., *Remote Sensing Digital Image Analysis: An Introduction,* Springer Verlag, 1986.

48. Roberts, D.A., Adams, J.B., Smith, M.O., "Predicted Distribution of Visible and Near-Infrared Radiant Flux Above and Below a Transmittant Leaf," *Remote Sensing of Environment,* 34, 1-17, 1990.